

Gerd Bauer

EasyKit

Eine Entwicklungsmethodik für das Rapid Engineering
miniaturisierter mechatronischer Systeme

EasyKit

**Eine Entwicklungsmethodik für das
Rapid Engineering miniaturisierter
mechatronischer Systeme**

Gerd Bauer



Universitätsverlag Ilmenau
2014

Impressum

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Angaben sind im Internet über <http://dnb.d-nb.de> abrufbar.

Diese Arbeit hat der Fakultät für Informatik und Automatisierung der Technischen Universität Ilmenau als Dissertation vorgelegen.

Tag der Einreichung: 28. März 2013

1. Gutachter: Univ.-Prof. Dr.-Ing. habil. Christoph Ament
(Technische Universität Ilmenau)

2. Gutachter: Univ.-Prof. Dr.-Ing. habil. Jens Müller
(Technische Universität Ilmenau)

3. Gutachter: Dr.-Ing. habil. Klaus Irrgang
(Temperaturmesstechnik Geraberg GmbH)

Tag der Verteidigung: 17. September 2013

Technische Universität Ilmenau/Universitätsbibliothek

Universitätsverlag Ilmenau

Postfach 10 05 65

98684 Ilmenau

www.tu-ilmenau.de/universitaetsverlag

Herstellung und Auslieferung

Verlagshaus Monsenstein und Vannerdat OHG

Am Hawerkamp 31

48155 Münster

www.mv-verlag.de

ISBN 978-3-86360-091-4 (Druckausgabe)

URN urn:nbn:de:gbv:ilm1-2013000509

Titelfoto: photocase.com

Kurzfassung

Die heutige Entwicklung mechatronischer Systeme ist meist durch einen sequentiellen Entwurfsprozess von Elektronik, Software und Mechanik gekennzeichnet. Gleichzeitig erfordert der domänenspezifische Entwurf häufig Entwicklungswerkzeuge in den einzelnen Fachgebieten, die einen hohen Grad an Expertenwissen erfordern und nur bedingt zueinander kompatibel sind.

Die Entwicklungsmethodik EasyKit hat das Ziel, durch eine Modularisierung auf Hard- und Softwareebene diesen Entwicklungsprozess zu vereinfachen. Dem Anwender werden hierfür Werkzeuge zur Verfügung gestellt, in die bereits umfangreiches Expertenwissen eingeflossen ist, so dass er mit einfachen Mitteln und Grundkenntnissen selbstständig die Hard- und Software für mechatronische Systeme entwickeln und aufbauen kann. Die vorliegende Arbeit beschreibt diese Werkzeuge und die damit verbundenen Technologien.

Vorgestellt werden verschiedene Möglichkeiten für eine lösbare Verbindung elektronischer Bausteine. Daneben wird ein Schnittstellenkonzept in Form eines außen liegenden Busses aufgezeigt, mit dem elektronische Funktionsträger einfach übereinander gestapelt werden können. Vorhandene Vorlagen für das Leiterplattenlayout und Testadapter unterstützen dieses Vorgehen.

Ein weiteres wesentliches Ziel dieser Arbeit ist es, den Entwicklungsprozess zu beschleunigen und wirtschaftlicher zu gestalten. Hierzu wurde ein Drei-Stufen-Modell entwickelt, das einen einfachen Übergang vom Funktionsmuster (Stufe 1) über die Kleinserie (Stufe 2) bis hin zum Serienprodukt (Stufe 3) ermöglicht. Vorliegende Funktionsblöcke in Hard- und Software führen zu einem schnellen Aufbau erster Funktionsmuster. Durch eine Wiederverwendung dieser Funktionsblöcke in allen drei Stufen kann die Überführung vom Funktionsmuster zum Serienprodukt stark beschleunigt werden. Dieses Konzept wurde ebenso auf die Software übertragen. Durch eine grafische Programmieroberfläche und verfügbare Funktionsbibliotheken sind Programme und lauffähige Codes deutlich leichter zu erzeugen.

An zwei praktischen Beispielen, einem Temperaturremesssystem und einem geregelten Pneumatikzylinder wird der komplette Entwurfsprozess dargestellt und Vorteile und Nachteile werden aufgezeigt. Ein Kostenvergleich zum konventionellen Vorgehen hebt die wirtschaftlichen Vorteile der Entwicklungsmethodik hervor.

Abstract

Today's development of mechatronic systems is usually characterized by a sequential design process of electronics, software and mechanics. Simultaneously, the domain-specific design often requires development tools in the various fields that require a high degree of expertise and that are only partially compatible.

The development methodology EasyKit aims to simplify the development process through modularization on hardware and software level. The tools for the user are provided for this purpose, they have been incorporated with extensive expertise, so that the user can independently develop with simple and basic knowledge of hardware and software and build up mechatronic systems. The present paper describes these tools and related technologies.

Different possibilities are introduced for a releasable connection of electronic components. In addition, an interface concept is shown in the form of an external bus which can be easily stacked with the electronic functional support to each other. Templates for printed circuit board layout and test adapters are supporting this approach.

Another important goal of this work is to accelerate the development process and make it more economical. For this purpose, a three-step model has been developed which allows an easy transition from functional models (level 1), small series (level 2) to the standard product (level 3). This functional blocks in hardware and software are leading to a rapid build up of the first functional prototype. To reuse this functional blocks in the three levels also can accelerate the way from a prototype to series production. This draught is also transferred to the software. With the use of a graphical user interface and available functional libraries, program and executable codes are very simply generateable.

The complete design process is shown in two practical examples, a temperature measuring system and a regulated pneumatic cylinder. Advantages and disadvantages of the development methodology are discussed. A cost comparison to the conventional approach shows the economic benefits of this development methodology.

Im Wald zwei Wege boten sich mir dar
und ich nahm den, der weniger betreten war.
Und dies änderte mein Leben.

Robert Frost (1874 – 1963)

Vorwort des Autors

Die vorliegende Arbeit entstand während meiner Tätigkeit als Geschäftsführer und Entwicklungsleiter der efm-systems GmbH. Die Entwicklung von miniaturisierten Auswertesystemen für die Mess- und Regelungstechnik stand dabei sowohl im persönlichen Fokus als auch im Interesse des Unternehmens. Wesentliche Ergebnisse des Projekts resultieren aus Arbeiten im Projekt EasyKit, das durch Fördermittel des Bundesministeriums für Bildung und Forschung unterstützt wurde. Hervorzuheben ist hierbei eine besonders enge und inspirierende Zusammenarbeit mit den Unternehmen digiraster Tetzner GmbH und Festo Didactic GmbH & CO. KG sowie den Universitäten TU München und TU Ilmenau.

Mein Dank gilt Herrn Professor Dr.-Ing. habil. Christoph Ament (TU Ilmenau) für die sehr gute Betreuung der Arbeit, angenehme Diskussionen und die Übernahme des Hauptberichts sowie Herrn Professor Dr.-Ing. Jens Müller für die Übernahme des Mitberichts. Ein besonderer Dank gilt Herrn Dr.-Ing. habil. Klaus Irrgang, der zum richtigen Zeitpunkt die Motivation für diese Arbeit wieder in mir geweckt hat und gleichzeitig ein Gutachten übernommen hat. Auch wenn die Arbeit am Ende das Thema der Temperaturerfassung nur am Rande gestreift hat, hoffe ich, dass wir mit weiteren gemeinsamen Arbeiten der Bedeutung dieses Themas noch ausreichend Rechnung tragen können.

Ein besonders herzlicher Dank gilt Frau Dipl.-Betriebswirtin (FH) Gabriele Schlenker für zahlreiche Korrekturrunden der Arbeit, hilfreiche Anregungen für die Wirtschaftlichkeitsbetrachtungen und viele Hinweise, einzelne Passagen noch einmal aus einem anderen Blickwinkel zu beleuchten. Ein ebenso herzlicher Dank gilt Frau Dipl.-Designerin (FH) Gesine Oberst für die Unterstützung beim Layout der Abbildungen. Ebenso möchte ich Herrn M. Sc. Thomas Koch für die Auffrischung meiner Kenntnisse in der Regelungstechnik danken.

Zurückblickend auf die Entwicklung von EasyKit möchte ich danken: Herrn Dipl.-Volksw. Klaus Zimmer für die Förderung im VDMA und die stetige Bereitschaft, das Thema in der Industrie bekannt zu machen, Herrn Professor Dr.-Ing. habil. Alois Knoll (TU München) und seinen Mitarbeitern für die Entwicklung der sehr innovativen Entwicklungssoftware EasyLab, Herrn Dr.-Ing. Reinhard Pittschellis (Festo Didactic GmbH & Co. KG) für den Mut, die Systeme industriell zu produzieren und das System weiter zu entwickeln und nicht zuletzt Herrn Dipl.-Ing. (FH) Rolf Eberbach (digiraster GmbH) für viele Ideen zur Aufbau- und Verbin-

dungstechnik und die Möglichkeit, diese Ideen unkompliziert gemeinsam umsetzen zu können.

Ein ganz besonderer Dank gilt an dieser Stelle Herrn Wolfgang Tetzner, der den Abschluss dieser Arbeit leider nicht mehr erleben durfte. Ihm verdanke ich viele neue Wege im Denken, im Herangehen an Arbeitsaufgaben und unkonventionellen Lösungsansätzen. Unsere Diskussionen fehlen mir.

Mein herzlicher Dank gilt meinen Eltern, Magdalene und Walter Bauer, die mich stets gefördert und mich auf meinem Bildungs- und Lebensweg best möglichst begleitet haben. Ganz besonders möchte ich am Ende aber meiner Frau Dorothée und meinen Kindern Kurt und Clara danken, die mir die Zeit und Ruhe gegeben haben, diese Arbeit erfolgreich beenden zu können.

Stuttgart, Dezember 2013

Inhaltsverzeichnis

1 Definitionen und Abkürzungen	14
1.1 Definitionen.....	14
1.2 Abkürzungen	17
2 Problemstellung und Zielsetzung	19
2.1 Problemstellung.....	19
2.2 Zielstellung.....	21
3 Analyse der Ausgangssituation.....	23
3.1 Entwicklungsmethodik für mechatronische Systeme	23
3.2 Anforderungen an den mikromechatronischen Entwicklungsprozess	27
3.3 Das Baukastensystem Modulare Mikrosysteme.....	28
3.4 Mikrotechnische Bausteine.....	30
3.5 Entwurfsalgorithmus für das Baukastensystem Modulare Mikrosysteme.....	34
3.6 Softwarewerkzeuge für die Entwicklung integrierter Systeme	36
4 Entwicklungsmethodik EasyKit.....	37
4.1 Charakterisierung der Rahmenbedingungen für die Entwicklungsmethodik.....	37
4.2 Struktur der Entwicklungsmethodik.....	39
4.3 Entwicklungsablauf.....	41
4.4 Entwicklungsaufgaben zur Umsetzung der Entwicklungsmethodik EasyKit.....	45
5 Technologiekonzept für die Hardware der Entwicklungsmethodik EasyKit....	46
5.1 3D-Aufbautechnik für Bausteine der Stufe 1	47
5.1.1 Aufbauform 1 – Funktionsträger mit Anschlüssen für steckbare Rahmen.....	48
5.1.2 Aufbauform 2 – Funktionsträger mit außenliegenden, lötbaren Anschlusskontakten.....	51
5.1.3 Aufbauform 3 – Funktionsträger in Starr/Flex-Technologie	55
5.1.4 Gegenüberstellung der Aufbauformen	58
5.2 Verbindungstechnik Stufe 1.....	60

5.2.1	Verbindung über Federmatrizen	61
5.2.2	Verbindung über Kontaktfolien.....	63
5.2.3	Verbindung über Matrixstecker	65
5.2.4	Gegenüberstellung der Verbindungsarten	67
5.3	Verbindungstechnik Stufe 2	71
6	Werkzeuge der Entwicklungsmethodik EasyKit	74
6.1	Schnittstellendefinition	74
6.1.1	Anpassung vorhandener Schnittstellen	74
6.1.2	Erweiterung um die elektromechanische Schnittstelle	76
6.2	Funktionsträger, Komponenten und Bausteine.....	78
6.2.1	Funktionsträger	78
6.2.2	Komponenten	79
6.2.3	Bausteine.....	80
6.3	Technologie und Entwicklungsprozess am Beispiel eines Beschleunigungsmesssystems.....	81
6.3.1	Funktionsstruktur und Teilfunktionen	82
6.3.2	Schaltungsentwicklung	85
6.3.3	Layout.....	86
6.3.4	Leiterplattenfertigung und -bestückung.....	88
6.3.5	Baustein- und Systemaufbau.....	90
6.3.6	Bausteindokumentation.....	93
6.4	Entwicklungs- und Testwerkzeuge.....	94
6.5	Programmierwerkzeug EasyLab	98
6.5.1	EasyLab – Die Idee.....	98
6.5.2	Grafische Programmieroberfläche	99
6.5.3	Funktionsbibliotheken.....	101
6.5.4	Programmausführung	102
6.5.5	Interaktives Debugging	104
6.5.6	Erweiterungsmodul.....	105
7	Anwendungsbeispiele.....	106
7.1	Temperaturmesssystem Pt100	106
7.1.1	Allgemeine Anforderungen.....	106

7.1.2	Funktionsstruktur und Teilfunktionen	107
7.1.3	Aufbau des Systems nach Stufe 1 der Entwicklungsmethodik ...	108
7.1.4	Vorgehen zur Ermittlung der Temperatur aus dem A/D-Wandlersignal	111
7.1.5	Erstellung der Firmware mittels konventionellem Ansatz.....	113
7.1.6	Erstellung der Firmware mittels EasyLab	116
7.1.7	Vergleich Software	120
7.1.8	Systemintegration in Stufe 1 der Entwicklungsmethodik.....	121
7.1.9	Aufbau des Systems nach Stufe 2 der Entwicklungsmethodik ...	122
7.1.10	Aufbau des Systems nach Stufe 3 der Entwicklungsmethodik ...	124
7.2	Pneumatikzylinder	126
7.2.1	Allgemeine Anforderungen	126
7.2.2	Modell des Regelkreises und Regelstrategie	127
7.2.3	Funktionsstruktur und Teilfunktionen	129
7.2.4	Aufbau des Systems nach Stufe 1 der Entwicklungsmethodik ...	130
7.2.5	Auslegung des Anschlussinterface	135
7.2.6	Aufbau des integrierten Pneumatikzylinders mit den EasyKit Bausteinen	137
7.2.7	Erstellung der Firmware mittels EasyLab	139
7.2.8	Zusammenfassung Pneumatikzylinder.....	145
8	Vergleich Wirtschaftlichkeit am Beispiel des Temperaturmesssystems	147
8.1	Vergleich der Entwicklungskosten	148
8.1.1	Entwicklungskosten nach konventionellem Entwicklungsvorgehen.....	148
8.1.2	Entwicklungskosten der Entwicklungsmethodik EasyKit bei komplett vorliegendem Bausteinsatz	150
8.1.3	Entwicklungskosten nach der Entwicklungsmethodik EasyKit bei einer Bausteinneuentwicklung.....	151
8.1.4	Zusammenfassung Entwicklungskosten	152
8.2	Vergleich Produktstückkosten.....	154
8.2.1	Produktstückkosten bei konventionellem Entwicklungsvorgehen.....	154
8.2.2	Produktstückkosten bei EasyKit	155
8.2.3	Vergleich der Produktstückkosten	157

9 Zusammenfassung.....	159
10 Abbildungs- und Tabellenverzeichnis.....	162
10.1 Abbildungsverzeichnis.....	162
10.2 Tabellenverzeichnis.....	167
11 Literaturverzeichnis.....	169

1 Definitionen und Abkürzungen

1.1 Definitionen

Aktor bzw. Aktuator

Aktoren dienen zur gezielten Beeinflussung von Zustandsgrößen in mechatronischen Systemen. Der Begriff Aktor umfasst alle Arten von Ausgabeelementen für Bewegungen und Kräfte.

A-Muster

Bezeichnet ein Funktionsmuster mit eingeschränktem Funktionsumfang hinsichtlich Kunden- und interner Spezifikationen, dient zur Bestätigung des Konzepts.

B-Muster

Bezeichnet einen Prototyp oder eine Kleinserienteil und ist geeignet für die Bestätigung des gesamten Funktionsumfang und der technischen Anforderungen.

C-Muster

Bezeichnet ein Vorserien- oder auch ein Serienteil, verfügt über alle technischen Spezifikationen und ist mit Methoden und Geräten für eine Großserienfertigung erprobt.

Code

Bezeichnung für das zu erstellende Programm oder Programmteile auf einem Mikrocontroller.

Eagle

Design Software zum Layout von Leiterplatten der Firma CadSoft.

EasyKit

Bezeichnung der Entwicklungsmethodik für miniaturisierte mechatronische Systeme, gleichzeitig auch der Name für ein vom BMBF gefördertes Entwicklungsprojekt „EasyKit – Innovative Entwicklungs- und Didaktikwerkzeuge für mechatronische Systeme“.

EasyLab

EasyLab ist ein Programmierwerkzeug mit grafischer Programmieroberfläche. Das Programm entstand im Rahmen des Projekts EasyKit durch die TU München.

Embedded Systems

In Deutsch auch eingebettete Systeme bezeichnet. Damit wird ein System mit einem Rechneranteil beschrieben (Mikrocontroller, Prozessor), der in einer Anwendung integriert ist und eigenständig Aufgaben der Signalverarbeitung, Steuerung und Regelung übernimmt.

Funktion

Die Funktion eines technischen Gebildes ist die für die Erfüllung eines Zwecks genutzte Eigenschaft, die Eingangsgröße E in die Ausgangsgröße A zu überführen [Krause 2004].

Funktionsstruktur

Die Funktionsstruktur beschreibt die Gesamtheit aller Teilfunktionen eines technischen Gebildes und deren Verknüpfungen.

Gesamtfunktion

Die Gesamtfunktion beschreibt die wesentlichen, d.h. die bestimmenden Funktion eines technischen Gebildes.

Hardware/Software Co-Design

Unter Hardware/Software Codesign wird der gemeinsame und vor allem parallele Entwurf von Hardware- und Softwareteilen eines Systems verstanden.

Mikrotechnische Bausteine

Im Rahmen der Entwicklung des Baukastensystems Match-X wurden die darin enthaltenen Bausteine mit der Bezeichnung Mikrotechnische Bausteine versehen.

Modbus

Kommunikationsprotokoll, das auf einer Master/Slave Architektur basiert [Modbus 2006].

Modulare Mikrosysteme

Die Aktivitäten zum Baukastensystem wurden ab 2001 unter dem Namen Modulare Mikrosystemtechnik zusammengefasst. Unter gleichen Namen entstand auch 2002 eine Arbeitsgemeinschaft im VDMA, die die Aktivitäten zum Baukastensystem koordiniert.

Rapid Engineering

Beschreibt eine Vorgehensweise zur schnellen Systementwicklung, speziell von Funktionsmustern und Prototypen.

Rapid Prototyping

Der Begriff definiert verschiedene Verfahren zur schnellen Herstellung von Musterteilen und Prototypen.

Stick-Slip-Effekt

Bezeichnet das Ruckgleiten von gegeneinander bewegten Körpern.

Teilfunktion

Teilfunktionen sind einzelne Bestandteile der Gesamtfunktion. Erst durch Zusammenwirken mehrerer Teilfunktionen kann eine Gesamtfunktion dargestellt werden.

V-Modell

Das V-Modell ist ein Vorgehensmodell, das ursprünglich für die Softwareentwicklung konzipiert wurde und anschließend auf die Mechatronikentwicklung übertragen und adaptiert wurde.

Zielformat

Der Begriff Zielformat wird häufig bei der Entwicklung eingebetteter Systeme verwendet, um den verwendeten Mikrocontroller und dessen Hardwareimplementierung zu beschreiben.

1.2 Abkürzungen

ADC

Analog/Digital-Wandler

AT

Arbeitstag

BGA

Ball Grid Array

BMBF

Bundesministerium für Bildung und Forschung

C, C++

Programmiersprachen

CAD

computer-aided design, rechnerunterstützter Entwurf

CPU

central processing unit, übliche Bezeichnung für die eingesetzten Mikrocontroller

DC

Direct current – Gleichstrom

EMV

Elektro Magnetische Verträglichkeit

FR4

Leiterplatte aus Epoxidharz und Glasfasergewebe

FS

Full Scale (Angabe bei Sensoren)

HW

Hardware

LTCC

Low Temperature Cofiring Ceramics

MID

Molded Interconnect Devices, Räumliche spritzgegossene Schaltungsträger

MLF-Package

Micro lead frame package

PCB

Printed Circuit Board

RS485

Schnittstellen-Standard für digitale leitungsgebundene, differentielle, serielle Datenübertragung

SMD

Surface Mount Devices

SMT

Surface Mount Technology

SW

Software

TB BGA

Top Bottom Ball Grid Array

VDMA

Verband Deutscher Maschinen- und Anlagenbau

µC

Mikrocontroller

2 Problemstellung und Zielsetzung

2.1 Problemstellung

Die Bedeutung der Elektronik und Software ist in allen Industriebranchen stark steigend. In Verbindung mit der Miniaturisierung von Komponenten und Systemen können so zunehmend neue und erweiterte Funktionalitäten, z.B. in Maschinen und Anlagen integriert werden. Der damit verbundene Einzug mechatronischer und mikromechatronischer Systeme verbindet gleichzeitig Forderungen nach einer hohen Zuverlässigkeit bei rauen Umgebungsbedingungen, hohen Belastungen und langer Lebensdauer [HOHWIELER 2012, ZEMAN 2006, JANSCHKE 2011].

Der Begriff der Mechatronik umfasst das interdisziplinäre Arbeiten in den Fachdisziplinen Mechanik, Elektronik und Software. Nach der VDI-Richtlinie 2206 [VDI2006 2003] wird als Mechatronik das synergetische Zusammenwirken der Fachdisziplinen Maschinenbau, Elektrotechnik und Informationstechnik beim Entwurf und der Herstellung industrieller Erzeugnisse sowie bei der Prozessgestaltung bezeichnet. Gebräuchlich ist darüber hinaus ebenfalls die Definition von Isermann [ISERMANN 1999]: „Mechatronik ist ein interdisziplinäres Gebiet, bei dem folgende Disziplinen zusammenwirken: Mechanische und mit ihnen gekoppelte Systeme, elektronische Systeme, Informationstechnik. Dabei ist das mechanische System im Hinblick auf die Funktionen dominierend. Es werden synergetische Effekte angestrebt, die mehr beinhalten als die reine Addition der Disziplinen.“ Heutige mechatronische Entwicklungen weisen jedoch einen immer größeren, zum Teil dominierenden Anteil von Elektronik und Software auf.

Die heutige Entwicklung mechatronischer Systeme ist durch einen sequentiellen Entwurfsablauf von Elektronik, Mechanik und Software gekennzeichnet. Während im Bereich der Mechanik verstärkt auf verfügbare Funktionsgruppen zurückgegriffen wird, zeichnet sich die Entwicklung im Bereich der Elektronik und insbesondere der Software durch die Erstellung spezifischer Schaltungen und Codes aus. Gleichzeitig führt der sequentielle Entwurfsablauf zu hohen Entwicklungszeiten und einem hohen Anpassungsaufwand in der Endphase der Entwicklung, wenn die einzelnen Teildisziplinen ineinander greifen müssen. Möchte man die Entwicklungszeiten verkürzen, um dadurch einen Wettbewerbsvorteil zu erreichen, so ist es nötig, die Entwurfsprozesse so weit wie möglich zu parallelisieren.

Die VDI2206 definiert eine Entwicklungsmethodik für mechatronische Systeme. Wichtiger Bestandteil dieser ist das V-Modell. Das V-Modell wurde ursprünglich für die Planung und Durchführung von IT-Projekten [HÖHN 2008] entwickelt und in der VDI2206 für die Anforderung der Entwicklung mechatronischer Systeme angepasst. Es beschreibt das generische Vorgehen beim Entwurf dieser mechatronischen Systeme. Zielstellung ist hierbei, bereits im Systementwurf einen domänenübergreifenden und somit ganzheitlichen Ansatz zu verfolgen und somit dem Problem der sequentiellen, getrennten Entwicklung in den Teildisziplinen entgegenzuwirken [THIELEMAN 2010].

Ein weiteres Problem ist, dass für die einzelnen Fachdisziplinen (Mechanik, Elektronik, Software) völlig unterschiedliche Entwurfswerkzeuge zur Verfügung stehen, die nicht untereinander kompatibel sind. Zur Bedienung dieser Werkzeuge ist häufig sehr fachspezifisches Expertenwissen erforderlich, so dass Unternehmen für ihre Produktentwicklung Experten in allen Teilgebieten der Mechatronik benötigen.

Die Trennung der verschiedenen Fachdisziplinen hat neben den technischen auch organisatorische Gründe: Damit das komplexe Gesamtsystem schließlich zuverlässig arbeitet, ist es bisher erforderlich schrittweise vorzugehen, so dass z.B. ein mechanischer Prototyp für die Verifikation der Elektronik verfügbar ist. Die vorliegende Elektronik ist wiederum erforderlich, um die Software abschließend entwickeln und testen zu können. Somit bestimmen diese Schritte häufig wieder den sequentiellen Entwicklungsablauf.

Die Entwicklung eines Produkts von der ersten Idee bis zur Serie ist meist durch drei Stufen gekennzeichnet. In Stufe 1 erfolgt die Entwicklung eines Funktionsmusters (A-Muster), dass die wesentliche technische Funktion des späteren mechatronischen Systems erfüllt. Stufe 2 ist die Prototyp- und Vorserienentwicklung (B-Muster). Hier werden sowohl die technischen Forderungen wie auch die baulichen Gegebenheiten und Feldanforderungen berücksichtigt. Erst in Stufe 3, der Serienentwicklung (C-Muster), erfolgt eine endgültige Überarbeitung des Systems unter Berücksichtigung der Wirtschaftlichkeit. Sowohl elektronische wie auch Softwarefunktionen werden während dieses Entwicklungsprozesses häufig umfassend geändert, was zu hohen Anpassungskosten und langen Entwicklungszeiten führt.

Speziell bei der Entwicklung mikromechatronischer Systeme kommen weitere Herausforderungen hinzu. Während in der Mikrotechnik Entwicklungen eher technologieorientiert sind, zeichnen sich klassische Mechatronikentwicklungen meist durch einen funktionsorientierten Ansatz aus. Ebenso ist die Stückzahl-/Kostenproblematik ein erhebliches Problem [BIERHALS 1999, SCHÜNEMANN 1999]. Die hohen Technologiekosten führen häufig zu hohen Fixkosten, die nur bei größeren Stückzahlen sinnvoll umgelegt werden können. Umgekehrt werden mikromechatronische Systeme häufig nur in kleineren und mittleren Stückzahlen benötigt.

Darüber hinaus ist der Zugang zu den Fertigungstechnologien, insbesondere im Bereich der Elektronik für viele Unternehmen ein schwer einzuschätzender Faktor und somit in vielen Fällen ein Hinderungsgrund für den Einstieg in die Mechatronik. Dabei führen insbesondere die Miniaturisierung und damit die immer weiter fortschreitende Spezialisierung der Fertigung zu einer Abhängigkeit von Entwicklungs- und Fertigungsdienstleistern [SCHMITT 2009].

Für viele kleine und mittlere Unternehmen ist der Zugang zu diesen mikromechatronischen Systemen und den dafür erforderlichen Technologien aus den genannten Gründen nur bedingt oder schwierig möglich. Die wesentlichsten Hemmnisse für diese Unternehmen sind zusammengefasst:

- Das erforderliche Expertenwissen bzw. die notwendigen Expertenwerkzeuge sind nicht bzw. nicht im erforderlichen Umfang vorhanden. Darüber hinaus sind diese Werkzeuge zumeist nicht oder nur bedingt untereinander kompatibel.
- Der Zugang zu den erforderlichen Technologien (z.B. Leiterplattenentwurf) und den dafür erforderlichen Werkzeugen ist nicht vorhanden.
- Die Entwicklungszeit und die damit verbundenen Entwicklungskosten sind in der Regel sehr hoch.

2.2 Zielstellung

Eine für die spezifischen Anforderungen der Entwicklung von miniaturisierten mechatronischen Systemen und von eingebetteten Systemen angepasste Entwicklungsmethodik liegt aktuell noch nicht vor. Diese Arbeit soll einen umfassenden Beitrag zur Entstehung einer solchen Entwicklungsmethodik leisten.

Als Ausgangspunkt werden die Anforderungen und Besonderheiten des Entwicklungsprozesses dieser Systeme analysiert und mit dem Stand der Technik abgeglichen. Das Konzept der Entwicklungsmethodik EasyKit und der dafür erforderlichen Werkzeuge wird erstellt.

Schwerpunkt der Arbeit ist die Entwicklung geeigneter Werkzeuge mit dem Schwerpunkt im Bereich der Hardware sowie die Schnittstellendefinition zu den parallel entstehenden Werkzeugen der Software.

Anhand von anwendungsnahen Beispielen soll die Entwicklungsmethodik und die begleitenden Werkzeuge auf ihre Tauglichkeit erprobt werden. Hierzu zählt ebenfalls eine Kostenbetrachtung unter Berücksichtigung der Entwicklungszeiten (Time to Market). Im Ergebnis liegt eine Gegenüberstellung der neuen Lösung hinsichtlich unterschiedlicher technischer und wirtschaftlicher Aspekte vor und die Vor- und Nachteile zu den bisherigen Methoden sind herausgearbeitet.



Abbildung 2.1: Vorgehen zur Ableitung der Entwicklungsmethodik und zur Entwicklung der dafür erforderlichen Werkzeuge

3 Analyse der Ausgangssituation

3.1 Entwicklungsmethodik für mechatronische Systeme

Modelle zur Konstruktionsmethodik finden im Maschinenbau bereits seit den 50er Jahren Anwendung [HANSEN 1955]. Es wird davon ausgegangen, dass durch eine Systematisierung des Prozesses die Lösung zielgerichtet ermittelt und somit sowohl die Qualität der Lösung gesteigert als auch die Entwicklungszeit verkürzt werden kann.

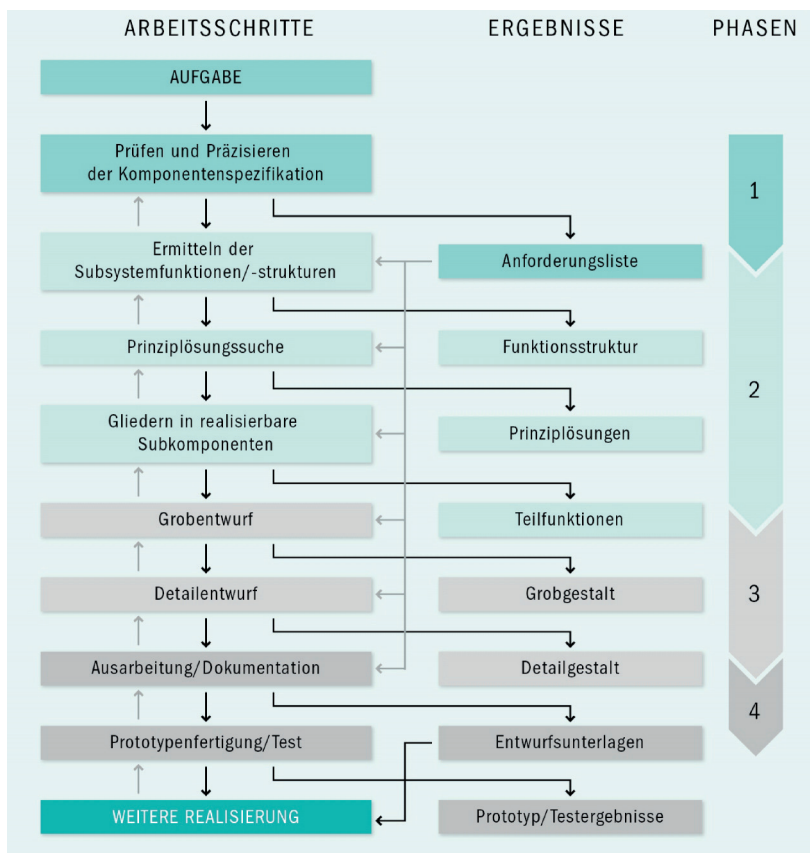


Abbildung 3.1: Erweitertes Phasenmodell der Konstruktionsmethodik

Dieser Ansatz wurde ständig weiter präzisiert und ist in unterschiedlichen Ausprägungsformen (z.B. Phasenmodell [PAHL 2005]) Grundlage für die heutige Konstruktionslehre. In Abbildung 3.1 ist ein erweitertes Phasenmodell dargestellt [KALLENBACH 2001], das auch in leicht abgewandelter Form für die Konstruktion mikrotechnischer Systeme aus dem Baukasten [SCHÜNEMANN 2003] Anwendung findet.

Während ursprünglich die Konstruktionslehre für rein mechanische Systeme vorgesehen war erfordert die Mechatronik eine Erweiterung des Ansatzes. In der VDI Richtlinie 2206 „Entwicklungsmethodik für mechatronische Systeme“ [VDI 2006 2003] werden das empfohlene Vorgehen und die Abläufe des mechatronischen Entwicklungsprozesses beschrieben.

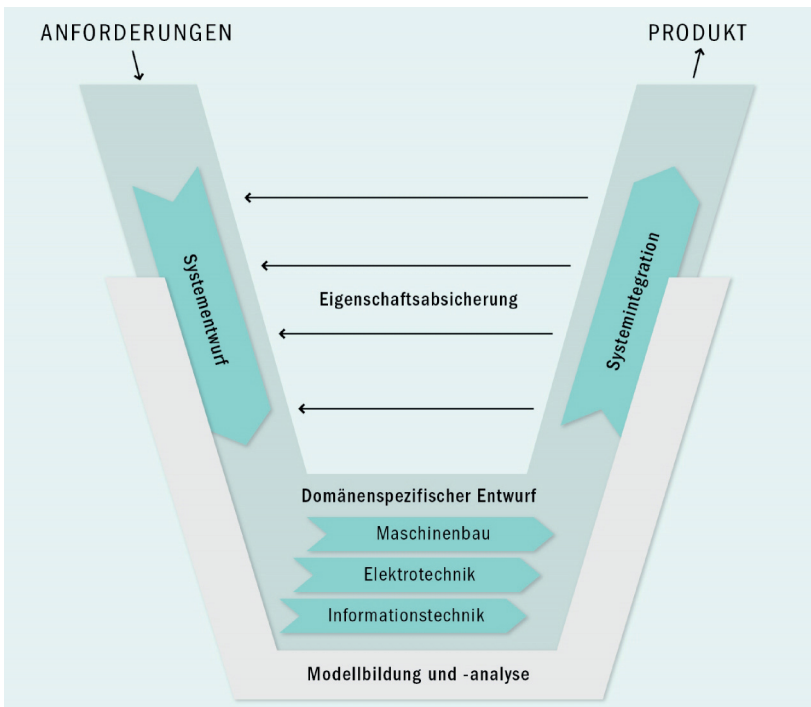


Abbildung 3.2: V-Modell als Makrozyklus nach VDI2206

Die Entwicklungsmethodik basiert im Wesentlichen auf drei Elementen, dem allgemeinen Problemlösungszyklus, dem V-Modell (siehe Abbildung 3.2) und vordefinierten Prozessbausteinen. Bei komplexeren mechatronischen Systemen ist dieser Makrozyklus mehrfach zu durchlaufen.

Im Bereich des domänenspezifischen Entwurfs ist entgegen den Empfehlungen der VDI2206 in der Regel ein sequentielles Vorgehen wie in Abbildung 3.3 dargestellt üblich. In einem ersten Schritt erfolgt die Konstruktion, anschließend folgt die Elektronik und erst im letzten Schritt die Entwicklung der Software auf den dann vorliegenden Hardwareplattformen von Mechanik und Elektronik. Den Wechselwirkungen zwischen den Teilprozessen wird mit diesem Vorgehen nur unzureichend Rechnung getragen.

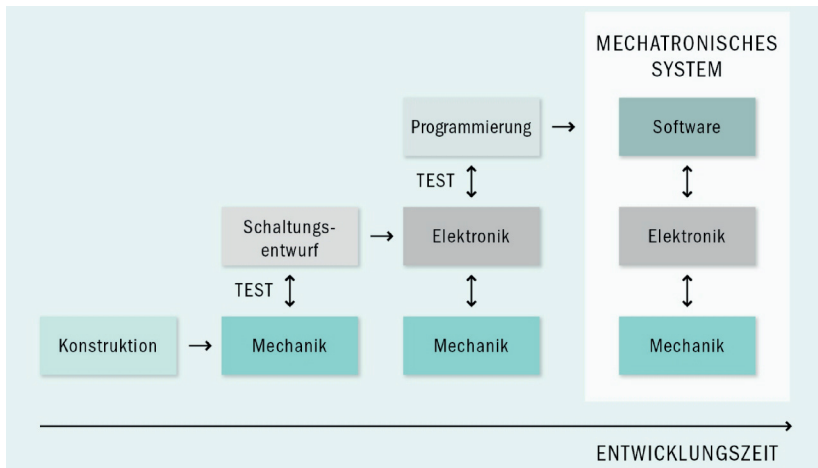


Abbildung 3.3: Sequentieller Entwurfsablauf der Mechatronik

Dabei führen umgekehrt die zunehmende Integration von Funktionen und die steigende Komplexität zu Wechselwirkungen zwischen den Teildisziplinen, die so früh wie möglich im Entwurf berücksichtigt werden müssen.

In den Teildisziplinen der Mechatronik gibt es unterschiedliche Entwurfswerkzeuge, die jeweils auf die Anforderungen dieser Teildisziplinen spezialisiert sind. Diese Entwurfswerkzeuge sind nicht oder nur bedingt untereinander kompatibel und erfordern zumeist umfassende Expertenkenntnisse. Die Unterstützung des in der VDI Richtlinie 2206 beschriebenen Vorgehens, insbesondere der Berück-

sichtigung der Wechselwirkungen zwischen den Teildisziplinen erfolgt durch diese Werkzeuge nicht oder nur unzureichend.

Gleichzeitig kann die Verwendung von verfügbaren und vorverarbeiteten Funktionsblöcken sowie standardisierter Normteile den Entwicklungsprozess stark beschleunigen. Jedoch ist in den unterschiedlichen Teildisziplinen der Mechatronik das Arbeiten mit Funktionsblöcken und Normteilen sehr unterschiedlich ausgeprägt.

In der Mechanik wird in der Regel ein Großteil einer Konstruktion aus vorhandenen Funktionsblöcken und standardisierten Normteilen realisiert. Dies wird von allen vorliegenden CAD-Werkzeugen umfassend unterstützt. Des Weiteren bieten die Anbieter von mechanischen Funktionsgruppen häufig die kompletten CAD-Daten ihrer Teile zur freien Verwendung mit an. Bereits in der Ausbildung wird das Arbeiten mit diesen so genannten Konstruktionselementen gelehrt und gefördert [KRAUSE 2004, STEINHILPER 2008].

In der Elektronik ist ein derartiges Vorgehen weit weniger verbreitet. Es gibt zwar für einzelne Bauelemente in den Datenblättern Vorschläge zur Beschaltung, aber komplette elektronische Funktionsblöcke, die z.B. in Form von Makros weitere Verwendung finden liegen firmenübergreifend nur vereinzelt vor. Darüber hinaus wird dieses Vorgehen auch nicht von den in der Elektronik üblichen Programmen zur Erstellung der Schaltpläne und Layouts unterstützt. Eine Datenübernahme zwischen den Programmen ist vielfach nur mit Einschränkungen möglich.

Ebenso ist bei der Erstellung von Programmen für Mikrocontroller die Verwendung von Funktionsblöcken und -bibliotheken, insbesondere firmenübergreifend, wenig verbreitet. In Internetforen sind zwar für eine Vielzahl von Funktionen Beispielcodes zu finden, die Qualität, Zuverlässigkeit und Richtigkeit dieses Codes ist jedoch nicht gewährleistet. Von den Anbietern der Mikrocontroller selbst gibt es ebenso vereinzelte Codes, aber auch hier ist die Zuverlässigkeit nicht gewährleistet. Darüber hinaus gibt es Unternehmen, die Bibliotheken für unterschiedliche Funktionen anbieten. Hier ist in der Regel die Zuverlässigkeit hoch, jedoch führt eine fehlende Schnittstellendefinition häufig zu einer aufwendigen Implementierung und Zusammenführung mit anderen Codeteilen. Für den Software Entwurfsprozess kommen verstärkt Werkzeuge auf den Markt, die eine funktionsgruppenorientierte Programmierung unterstützen. Jedoch hängt der Einsatz stark von den Ressourcen des Endgerätes ab. Insbesondere in Ver-

bindung mit 8- und 16-bit Mikrocontrollern stehen derartige Werkzeuge noch nicht zur Verfügung. Daher steht in diesen Bereich der Einsatz von Funktionsgruppen erst am Anfang. In der Praxis wird der Quellcode für unterschiedliche Applikationen komplett neu erstellt. Die Wiederverwendung einzelner Funktionsgruppen findet nur firmenintern in kleinem Umfang statt.

3.2 Anforderungen an den mikromechatronischen Entwicklungsprozess

Die Mikromechatronik ist ein kleiner Teilbereich der Mechatronik mit stark zunehmender Bedeutung. Der Trend zur weiteren Miniaturisierung und der Einsatz miniaturisierter intelligenter Systeme zur Erhöhung der Funktionalität von Komponenten, Maschinen und Anlagen in Branchen wie dem Maschinen- und Anlagenbau werden vielfach als Schlüssel für einen nachhaltigen Erfolg und dem entscheidenden Wettbewerbsvorteil gesehen [KÜHNEL 2010]. Dieser Trend zur weiteren Miniaturisierung und zu einer stetig steigenden Funktionalität führt zu kontinuierlich hohen Steigerungsraten dieser Teildisziplin. Gleichzeitig ist die Mikromechatronik häufig gekennzeichnet durch einen erhöhten Anteil der Bereiche Elektronik und Software. So ist auch die Entwicklung eingebetteter Systeme (Embedded Systems) oder von kleinen kompakten eigenständig fungierenden Sensor-/Aktorsystemen (smart systems integration) häufig den Bereich der Mikromechatronik zuzuordnen. Dabei steht der mikromechatronische Entwicklungsprozess an der Schnittstelle zwischen zwei konträren Vorgehen, dem klassischen Entwurfsprozess der Mechatronik wie in Kapitel 3.1 beschrieben, der stark funktionsorientiert ist und dem Entwurfsprozess der Mikrotechnik, der seine Ursprünge in der Mikroelektronik hat und sehr stark technologieorientiert ist [ZÖPPIG 2005].

Darüber hinaus unterliegt die Mikromechatronik einer ausgeprägten Stückzahl-/Kostenproblematik. Die meist hohen Entwicklungskosten für mikromechatronische Systeme sind in vielen Fällen verknüpft mit hohen Initialkosten für die Produktion dieser Systeme. Dies führt in vielen Fällen dazu, dass sich diese Systeme nur bei großen Stückzahlen rechnen [GRIMME1996]. Umgekehrt existiert ein großes Anwendungspotenzial für derartige Systeme in einer Branche wie der Investitionsgüterindustrie, bei der jedoch üblicherweise nur ein Bedarf an kleineren und mittleren Stückzahlen besteht [VAN DE VENN 2006]. Dieser Widerspruch wird als Haupthemmnis für einen Einsatz und die Verfügbarkeit mikromechatronischer Systeme zu marktakzeptablen Preisen gesehen

[WESTKÄMPER 1997]). Gleichzeitig liegt in dieser Branche häufig in einzelnen Unternehmen nicht das Expertenwissen in den unterschiedlichen Bereichen der Mechatronik vor, um diese Systeme im eigenen Haus zu entwickeln.

Als Lösungsansatz zur Überwindung dieser Problematik wurden in Vorarbeiten die Modularisierung und der Aufbau eines Baukastensystems empfohlen [BAUER 2001A, BAUER 2004A, SCHÜNEMANN2003].

3.3 Das Baukastensystem Modulare Mikrosysteme

Auf Initiative des VDMA wurde ein Projekt zur Ermittlung der Innovationspotentiale der Mikrosystemtechnik im Maschinenbau gestartet [SCHÜNEMANN 1996], die in einem Leitprojekt des BMBF, „Mikrosysteme für den Maschinen und Anlagenbau“, von 1996 bis 1998 näher untersucht wurden [VDMA 1998]. Im Ergebnis dieses Projekts wurde ein Baukastensystem der Mikrosystemtechnik vorgeschlagen [SCHÜNEMANN 1998B, GROßER 1999]. Dieses Baukastensystem basiert auf der Bereitstellung von Funktionsgruppen, die in unterschiedlichen Systemen und Aufbauvarianten Anwendung finden und so in größeren Stückzahlen hergestellt werden können. Gleichzeitig führt die Verfügbarkeit dieser Funktionsgruppen wie in Abbildung 3.4 dargestellt zu einer hohen Variationsvielfalt und zur einfachen Realisierung von gleichartigen Systemen mit variierenden Ausstattungsmerkmalen (z.B. gleiche Sensorfunktion mit unterschiedlicher Buskommunikation).

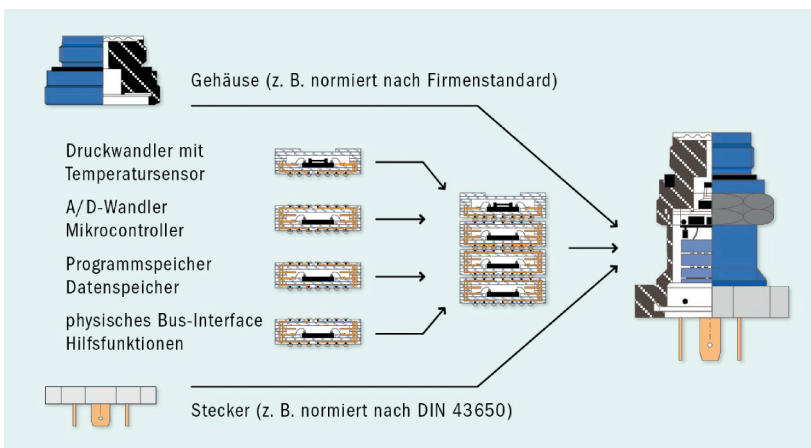


Abbildung 3.4: Baukastenkonzept Modulare Mikrosysteme für den Maschinen- und Anlagenbau

Als Lösungsansatz für die Entwicklung und Herstellung von kostengünstigen Systemen wurde somit die Modularisierung vorgeschlagen [BAUER 2001B, SCHÜ-NEMANN 2001]. Durch Kombinationen aus definierten Baugruppen eines mikro-technischen Baukastens können unterschiedliche miniaturisierte Systeme kostengünstig hergestellt werden. Zur Unterstützung dieses Ansatzes wurde ein neuartiges 3D-Aufbaukonzept konzipiert und in ersten Funktionsmustern realisiert, das Top-Bottom-BGA [AMIRI JAM 1998, LEUTENBAUER 1998].

Das in der Industrie auf großes Interesse stoßende Konzept des mikrotechnischen Baukastensystems führte zu einem weiteren Leitprojekt des BMBF, das unter der Bezeichnung „Industrieplattform Modulare Mikrosysteme“ von 1999-2001 lief und den technischen und ökonomischen Nachweis der Machbarkeit zum Ziel hatte. Die Aktivitäten zum Baukastensystem wurden 2001 unter der Bezeichnung Modulare Mikrosysteme zusammengefasst und von einer Arbeitsgemeinschaft des VDMA koordiniert. Zielstellung war die Etablierung eines herstellerübergreifenden Baukastens. Voraussetzung hierfür waren standardisierte, offene Schnittstellen zwischen den Bausteinen. Daher wurden die Schnittstellen zwischen den Bausteinen entwicklungsbegleitend standardisiert [VDMA 2003]. Da das Baukastensystem alle Teilbereiche der Mikrosystemtechnik abdecken sollte, erfolgte die Standardisierung der geometrischen, elektrischen, fluidischen, optischen, mechanischen und thermischen Schnittstelle. Bei der Überführung in praktische Anwendungen zeigte sich jedoch, dass nur die geometrische, elektrische und fluidische Schnittstelle praktikabel waren. Die optische und die mechanische Schnittstelle stießen aufgrund der aus den Anwendungen resultierenden Anforderungen und der vorgegebenen Einbaubedingungen immer wieder auf technische Grenzen.

Ab 2002 fanden im Rahmen mehrerer öffentlich geförderter Projekte Entwicklungen zu konkreten Bausteinen und Systemen statt¹. Dabei wurden mehrere Probleme offensichtlich:

- Die vorgeschlagene Aufbau- und Verbindungstechnik erwies sich als sehr kostenintensiv, Kostenvorteile gegenüber konventionellen Aufbauten konnten auch bei kleineren Stückzahlen nicht erreicht werden.

¹ Eine Auflistung der öffentlich geförderten Projekte ist aufgrund der großen Anzahl hier nicht möglich. Eine Übersicht zu den Projekten und den Projektergebnissen ist unter <http://www.mstonline.de/foerderung/projektliste> einfach möglich.

- Die Zuverlässigkeit der aufgebauten Bausteine und Systeme entsprach nicht den Erwartungen.
- Die Testbarkeit war nicht gewährleistet.
- Entwicklungswerkzeuge für den Bausteinentwurf und die Fertigung fehlten völlig.
- Der Softwareentwurf wurde bei den Überlegungen zur Entwicklung des Baukastensystems nicht berücksichtigt.
- Die Anzahl der am Markt verfügbaren Bausteine war zu gering.

Viele dieser Entwicklungen wurden daher nicht weiter verfolgt und eingestellt.

3.4 Mikrotechnische Bausteine

Für das Baukastensystem wurde, wie bereits kurz aufgeführt, ein neuartiges Packagingkonzept, das Top-Bottom-BGA entwickelt [LEUTENBAUER 1997B]. Hierbei handelt es sich um ein geschlossenes Gehäuse mit elektrischen Schnittstellen an der Ober- und Unterseite des Gehäuses. Diese Schnittstellen ermöglichen ein Stapeln dieser Gehäuse, die im Weiteren auch als mikrotechnische Bausteine bezeichnet werden.

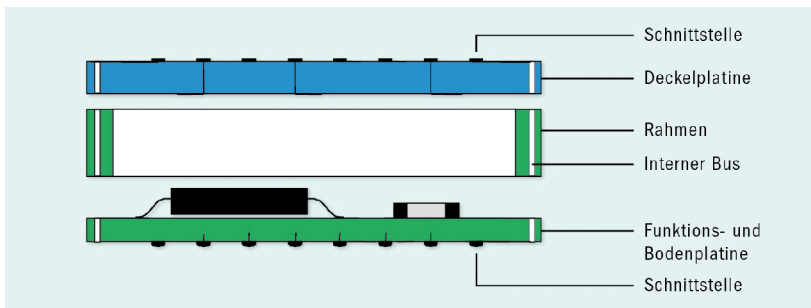


Abbildung 3.5: Top-Bottom-BGA

Der Baustein besteht in der ursprünglichen Konzeption aus 3 Komponenten, der kombinierten Funktions- und Bodenplatte, dem Rahmen und der Deckelplatte. Die kombinierte Funktions- und Bodenplatte weist auf einer Seite die Schnittstelle zum Anschluss eines weiteren Bausteins auf, auf der anderen Seite sind Bestückungsmöglichkeiten für die Funktionskomponenten. Gleichzeitig wird

diese Platine auch für die elektrische Umverdrahtung genutzt. Daher ist für den Aufbau in aller Regel ein mehrlagiger Aufbau erforderlich. Der Rahmen hat zwei wesentliche Funktion, die mechanische Überbrückung der integrierten Funktionskomponenten und die Führung des internen Bussystems des Bausteins. Die wesentliche Funktion der Deckelplatine ist wiederum die Bereitstellung der äußeren Schnittstellen für den Anschluss weiterer Bausteine.

Die Schnittstellen der Bausteine wurden entwicklungsbegleitend im VDMA Einheitsblatt 66305 standardisiert. Die wesentlichsten Standards sind hier kurz zusammengefasst: Die Bausteine haben eine quadratische Grundfläche. Diese kann in den Abmessungen 5mm x 5mm bis 25mm x 25mm in einer Abstufung von 2,5mm variieren. Die Bausteingröße von 12,5mm x 12,5mm wurde als Vorzugsgröße definiert.

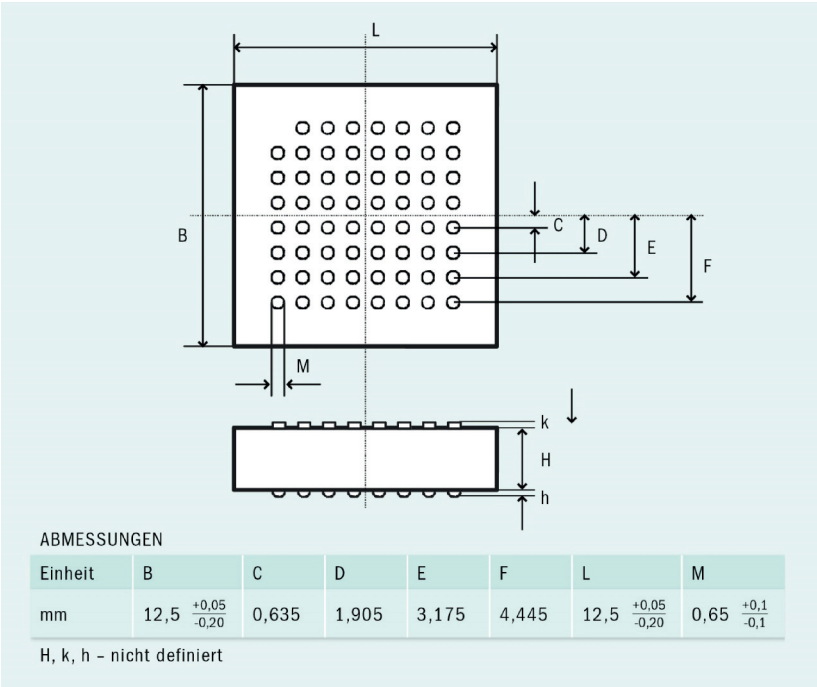


Abbildung 3.6: Geometrisch/elektrische Schnittstelle der Bausteingröße 12,5mm x 12,5mm

Die elektrische Schnittstelle ist in Form eines Grid Arrays mit einem Raster von 1,27mm auf der Ober- und Unterseite des Bausteins angeordnet. Im Fall eines Bausteins der Größe 12,5mm sind 63 Anschlüsse im Raster von 8x8-1 angeordnet (siehe Abbildung 3.6).

Neben der Definition der Lage und Abmessungen dieser Anschlüsse wurde auch deren Belegung definiert (siehe Abbildung 3.7) [VDMA2003].

Des Weiteren wurden neben den geometrischen und elektrischen Schnittstellen auch die fluidische, die optische, die mechanische und die thermische Schnittstelle definiert. Diese umfassende Definition sollte dem Charakter eines universellen Baukastensystems der Mikrosystemtechnik Rechnung tragen. Zum Teil wurden auch Bausteine mit einer fluidischen Schnittstelle aufgebaut [AMIRI JAM 2004, SCHINDLER 2005] und in kompletten Systemaufbauten getestet.

	5	6	7	8	9	10	11	12
E		MS1	MS2	MS3	MS4	RB6	RB7	GD2
F	PRG	RES	CS	RC0	NC	SD0	SDI	UD2
G	RC1	INT	GD1	X	X	UD1	SCK	RC2
H	X	CANL	TXD	X	X	RXD	CANH	X
J	NC	NC	X	X	X	X	X	NC
K	RA0	NC	UA1	PT1	PT2	GA1	NC	RA3
L	RA1	NC	URP	PT3	PT4	URM	NC	RA4
M	RA2	NC	GA2	NC	NC	UA2	NC	RA5

Abbildung 3.7: Elektrische Schnittstelle (Anschlussbelegung)

Für die ersten Entwicklungen fand eine Konzentration auf die geometrisch-elektrische Schnittstelle statt. Die erste Generation der mikrotechnischen Bausteine wurde in LTCC-Technologie hergestellt. Hierbei wurde bereits während des Herstellungsprozesses der Rahmen auf die Bodenplatte laminiert und

anschließend gemeinsam verpresst. In den so entstandenen Grundkörper wurden die Funktionskomponenten bestückt und verlötet. Im letzten Arbeitsschritt wurde die Deckelplatine aufgesetzt und wahlweise durch Verlöten oder Verkleben mit dem Grundkörper verbunden [LEUTENBAUER 1997B]. Die wesentlichen Nachteile an diesem Fertigungsablauf waren die geringe Verfügbarkeit dieser Fertigungstechnologie und die sehr hohen Kosten des Aufbaus in LTCC-Technologie bei gleichzeitig sehr kleinen Stückzahlen.

Daher wurde in der zweiten Generation der Aufbau der mikrotechnischen Bausteine leicht angepasst und der Fertigungsablauf auf die Anforderungen der Leiterplattentechnologie adaptiert [AMIRI JAM 2000]. Die Bodenplatine ist eine mehrlagig (typisch 4-6 Lagen) ausgeführte FR4-Leiterplatte, die auf der Oberseite Träger für die Funktionskomponenten ist und auf der Unterseite die standardisierte Schnittstelle aufweist. Der Rahmen ist aus dem gleichen Basismaterial. Der Innenraum wird durch Fräsen freigeräumt. Der Rahmen enthält umlaufend Durchkontaktierungen für den internen Bus. Die Höhe des Rahmens ist frei gestaltbar. Die Deckelplatine wiederum ist ein zweilagiger Aufbau, der die Umverdrahtung vom internen Bussystem auf die externen Schnittstellen des Bausteins enthält. Die Funktionskomponenten werden mit Standardbestückungstechnologien auf die Bodenplatine aufgebracht und verlötet. Anschließend wird der Rahmen und Deckel aufgesetzt und ebenfalls verlötet. Diese Technologie erwies sich in der Handhabung und der Verfügbarkeit als deutlich einfacher als die LTCC-Technologie. Allerdings traten bei den ersten Mustern noch Probleme hinsichtlich der Zuverlässigkeit und der Testbarkeit der Bausteine auf.

Eine weitere wesentliche Anforderung an den Baustein war, nicht nur ein Packaging für einzelne elektrische oder elektromechanische Funktionskomponenten darzustellen, sondern eine komplett eigenständige Funktionsgruppe zu repräsentieren. Da hierfür komplette Baugruppen in den Baustein integriert werden müssen, führt dies zu hohen Packungsdichten im Baustein. In vielen Fällen hat der Bauraum jedoch nicht dafür ausgereicht. Daher wurde in einer weiteren Generation auch die Unterseite der Deckelplatine als Bestückungsfläche genutzt. Dieser Aufbau war jedoch mit dem Nachteil verbunden, dass viele nur innerhalb einer Baugruppe genutzte Signale über den internen Bus geführt werden mussten und dieser so schnell an die Kapazitätsgrenzen stieß.

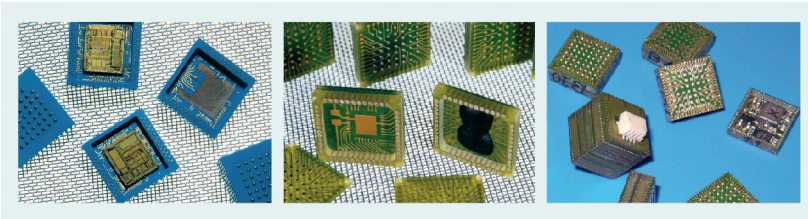


Abbildung 3.8: Links: Bausteine der ersten Generation (LTCC-Technologie), Mitte: Bausteine der zweiten Generation (FR4-Technologie mit geklebtem Deckel), rechts: Bausteine der dritten Generation (FR4-Technologie mit gelötetem Deckel)

Zum Abschluss der Arbeiten am mikrotechnischen Baukastensystem lagen unterschiedliche erfolgreich getestete Aufbauvarianten vor [BAUER 2009B]. Jedoch verhinderten der noch deutlich zu hohe Preis dieser Bausteine, die geringe Verfügbarkeit und Probleme in der Zuverlässigkeit eine weitere Verbreitung dieses Systems. Ebenso wirkten sich fehlende Werkzeuge für die Entwicklung und den Test wie auch für die Erstellung der Software negativ aus.

3.5 Entwurfsalgorithmus für das Baukastensystem Modulare Mikrosysteme

Für den systematischen Entwurf von mikrotechnischen Systemen aus dem Baukasten wurde ein Entwurfsalgorithmus (siehe Abbildung 3.9) erstellt. Dieser basiert auf der VDI2221 [VDI2221 1993] und wurde für die Anforderungen des Baukastensystems angepasst [SCHÜNEMANN 1998A].

Dabei wird nach der Präzisierung der Anforderung und der Aufstellung der Gesamtfunktion und der dazugehörigen Funktionsstruktur diese in die Teilfunktionen zerlegt. Diesen Teilfunktionen werden wiederum soweit vorhanden Bausteinen aus dem Baukastensystem zugeordnet und so ein Satz optimalverdächtiger Bausteine ermittelt. Dieser Satz wird in seinem Zusammenwirken überprüft und bei Eignung in ein Gesamtsystem überführt.

Der Ablauf ist insbesondere geeignet für die Auswahl elektronischer und elektromechanischer Funktionen und Bausteine. Das Zusammenwirken mit der Software wurde in diesem Entwurfsalgorithmus nicht berücksichtigt.

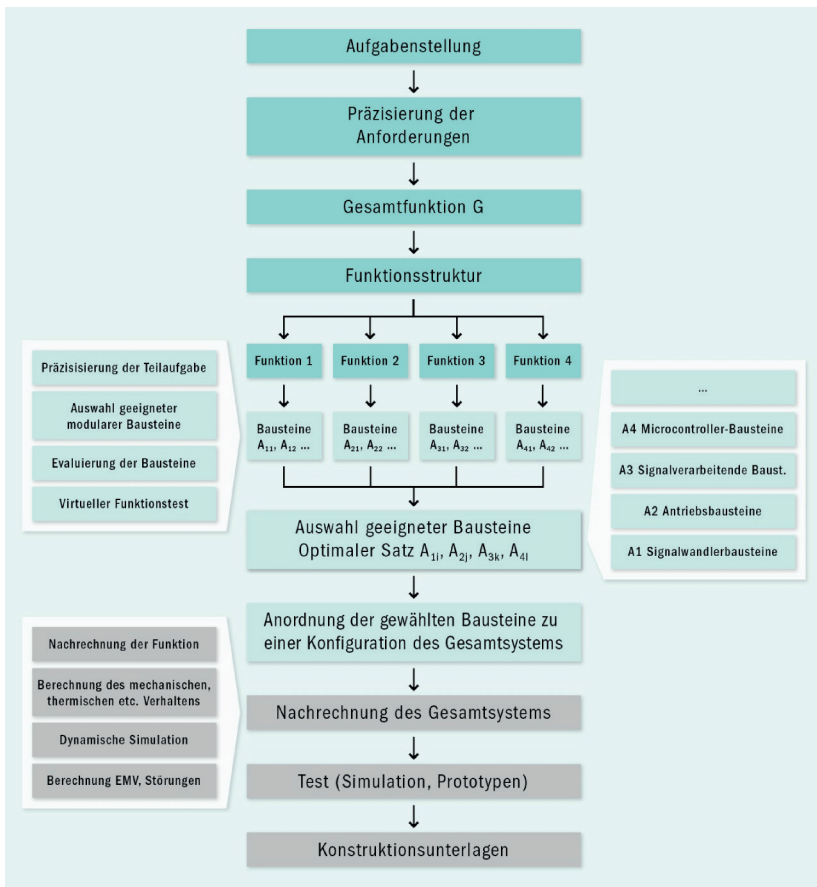


Abbildung 3.9: Entwurfsalgorithmus des Baukastensystems Modulare Mikrosysteme

Zusammenfassend ist für die Kapitel 2.2 bis 2.5 jedoch festzustellen, dass dem Baukastensystem die ganzheitliche Betrachtung fehlte. Das Zusammenwirken von Mechanik, Elektronik und Software, die Einbindung in eine begleitende Entwicklungsmethodik und somit die Senkung der Eintrittsschwellen in ein solches System wurden nicht ausreichend berücksichtigt. Aus den genannten Gründen konnte sich das Baukastensystem, das ab 2004 in Teilen am Markt verfügbar war [BAUER 2004B], nicht durchsetzen.

3.6 Softwarewerkzeuge für die Entwicklung integrierter Systeme

Für die Entwicklung mechatronischer Systeme werden zunehmend Werkzeuge für die Erstellung von Software bereit gestellt, die auch ohne tiefgehendes Expertenwissen in den Programmiersprachen einen Zugang zur eigenständigen Programmierung von Systemen ermöglichen [SCHÄFER 2009, MOSER 2007]. Diese Werkzeuge verfügen häufig über eine grafisch orientierte Entwicklungsumgebung und eine integrierte Code-Generierung. Dass solche Konzepte ein enormes Potenzial bei einem großen Anwenderkreis haben, zeigt der große Erfolg der beiden Systeme Matlab-Simulink und LabView. Erstgenanntes System ist eine sehr umfangreiche, historisch gewachsene, grafikbasierte, datenfluss-orientierte Simulationsumgebung mit der Möglichkeit zur Einbindung von Hardware-Komponenten in den Simulationsablauf.

LabView verfolgt den gleichen Ansatz der Datenfluss-Orientierung (der das ingenieurmäßige Denken ideal abbildet) und wurde zunächst für die Laborautomatisierung entwickelt – auch heute ein wichtiges Einsatzfeld für dieses System. Wie bei Matlab steht aber auch hier die Unterstützung des Software-Entwicklungsprozesses für Echtzeit-Zielsysteme nicht im Vordergrund. So benötigt LabView zur Ausführungszeit einen Host-Rechner (z.B. PC unter Windows). Zwischenzeitlich ist auch die Code-Generierung für 32-bit und 16-bit Prozessoren möglich. Eine weitere Skalierung hin zu kleineren Prozessoren ist jedoch kurz- und mittelfristig nicht zu erwarten.

Umgekehrt besteht aber die Forderung, für kleine Prozessoren, die insbesondere für die Sensorauswertung oder für Sensornetzwerke eingesetzt werden, ebenfalls eine Entwicklungsunterstützung bereitzustellen. Aus Softwaresicht ist der Bereich Sensornetzwerke und die damit verbundene Erstellung der Software als Bereich mit großem Entwicklungspotenzial einzustufen. Hiervon zeugen eine Vielzahl laufender Forschungs- und Entwicklungsprojekte in diesem Bereich [ZHAO 2012, DZIENGEL 2012]². Die einzelnen Systeme werden manuell und sehr anwendungsbezogen implementiert, eine Wiederverwendung von Codeteilen findet praktisch nicht statt.

² Eine umfangreiche Auflistung und Präsentation geförderter Projekte zu diesem Thema unter: <http://www.vdivde-it.de/anmeldung/statusmeeting>

4 Entwicklungsmethodik EasyKit

4.1 Charakterisierung der Rahmenbedingungen für die Entwicklungsmethodik

Durch die starke Technologieorientierung und das große erforderliche Expertenwissen in den Teilgebieten der Mikromechatronik führt der übliche Entwicklungsablauf zu sehr langen Entwicklungszeiten verbunden mit sehr hohen Entwicklungskosten. Bereits in der Entwurfsphase werden häufig Entscheidungen getroffen, die später nur mit großem Aufwand zu revidieren sind und sich domänenübergreifend auswirken. Wesentlich hierbei ist die Wechselwirkung zwischen Hard- und Software, die meist erst mit vorliegender Hardware in einer späten Phase der Entwicklung getestet werden kann. Dann auftretende Probleme sind häufig nur durch einen Wechsel in der Plattform (CPU) oder in einer aufwendigeren Gestaltung des Codes zu beseitigen, beides mit den Folgen höherer Kosten und vor allem deutlich längerer Entwicklungszeiten.

Daher ist es erforderlich, den Entwicklungsablauf, insbesondere in der Wechselwirkung der Hard- und Software flexibler zu gestalten und bereits in einer sehr frühen Entwicklungsphase die Anforderungen der Software stärker einzubinden. Ziel ist daher bereits in der frühen Entwurfsphase eine Verknüpfung von Hard- und Software, die ein iteratives Vorgehen ermöglicht, das Hardware/Software-Codesign [SCHAUMONT 2010]. So können z.B. mit vorliegender Hardware zentrale Fragestellungen zur Software (z.B. Speicherbedarf und Ressourcen, parallel zu verarbeitende Prozesse) überprüft werden. Voraussetzung hierfür ist aber das Vorliegen von Funktionsblöcken, die zumindest in den Grundzügen der Beschaltung dem späteren Einsatz Rechnung tragen. Parallel ist es aber auch in der Software sinnvoll, wiederkehrende Funktionen nicht neu zu programmieren, sondern diese aus Bibliotheken wieder zu verwenden.

Somit ist ein erster zentraler Lösungsansatz der Einsatz von vordefinierten Funktionen und Funktionsblöcken. Diese können prinzipiell in unterschiedlichsten Formaten vorliegen, z.B. als Hardware (elektronischer Funktionsbaustein), als Schaltungsbibliothek und/oder Layoutbibliothek, die schnell umgesetzt werden können oder als Softwarefunktionsblock. Voraussetzung für einen Einsatz ist jedoch, dass die Schnittstellen dieser Funktionsblöcke beschrieben und so definiert sind, dass eine Kompatibilität zwischen verschiedenen Funktionsblöcken sicher gestellt ist.

Der Wechselwirkung zwischen Hardware und Software ist Rechnung zu tragen, indem neben allgemeinen Softwarefunktionsblöcken (z.B. mathematische Bibliotheken, Regler) auch solche für speziell vorliegende Hardware bereits zur Verfügung steht. Gleichzeitig sollte das Werkzeug zur Erstellung der Software eine Simulation von Programmen ermöglichen, die unter Berücksichtigung dieser Hardware eine schnelle Abschätzung der Umsetzbarkeit hinsichtlich Leistungsfähigkeit und Ressourcen der verwendeten Controllerplattform erlaubt und so das Hardware/Software-Codesign unterstützt.

Ein weiterer Lösungsansatz ist, den Entwicklungsablauf durch begleitende Werkzeuge zu unterstützen. Der Einsatz von Funktionsblöcken in Hard- und Software mit definierten Schnittstellen führt zwingend zu Schnittstellenbeschreibungen und zu einer Präferenz an Technologien und Werkzeugen. Gleichzeitig kann das Vorgehen so verallgemeinert und beschrieben werden, dass auch für einen Dritten der Einsatz einfach möglich wird, mit geringeren Eintrittsschwellen verbunden ist und durch den Zugriff auf vorliegende Werkzeuge unterstützt wird.

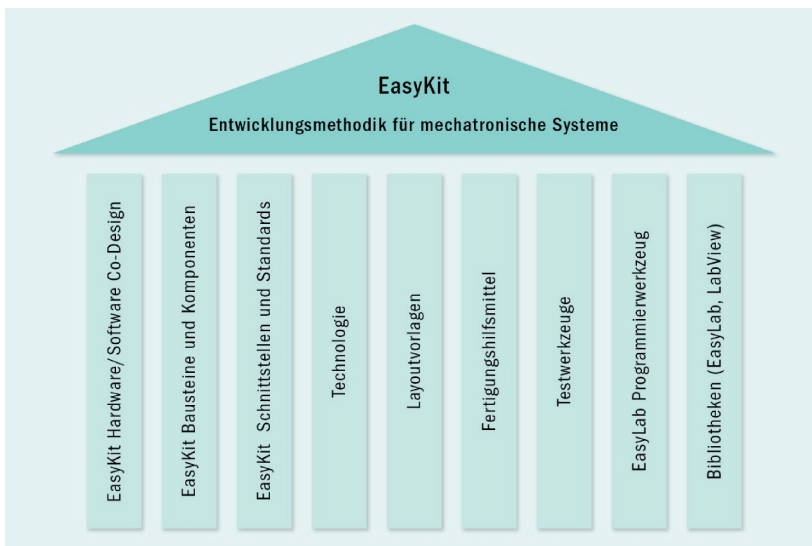


Abbildung 4.1: Entwicklungsmethodik und -werkzeuge

Werkzeuge für die Entwicklung, die Fertigung und den Test können herstellerübergreifend zur Verfügung gestellt werden und somit den Entwicklungsprozess bei geringeren Kosten verkürzen. Diese Werkzeuge müssen dann in den unterschiedlichen Stufen des Entwurfs- und Entwicklungsprozesses zur Verfügung stehen.

Die Zusammenführung dieser Lösungsansätze führt zu einer Entwicklungsmethodik, die diesen Entwurfsablauf definiert. Diese Entwicklungsmethodik muss den Ablauf und die Werkzeuge beschreiben, die dem Anwender einen einfachen und verständlichen Zugriff ermöglichen.

Auf Grundlage dieser Anforderungen wurde die Entwicklungsmethodik EasyKit [BAUER 2009A, BARNER 2010, PITTSHELLIS 2010] konzipiert.

4.2 Struktur der Entwicklungsmethodik

Die Entwicklungsmethodik EasyKit definiert das bildliche Gebäude, unter dem der Ablauf, die Werkzeuge und die Schnittstellen eines Vorgehens zur schnellen und kostengünstigen Entwicklung miniaturisierter mechatronischer Systeme zusammengefasst werden. Alle Teilbereiche sollen der Forderung Rechnung tragen, sehr spezielles Expertenwissen zu reduzieren. In einem ersten Konzept werden folgende unterstützende Säulen definiert.

EasyKit Hardware/Software Codesign

Beschrieben wird der Entwurfsablauf, die verschiedenen Stufen vom Funktionsmuster zum Serienprodukt sowie die zur Verfügung stehenden Hilfsmittel.

Bausteine und Komponenten

Unter diesen Bereich fällt insbesondere die vorliegende Hardware, die in Form von Bausteinen oder Komponenten für den Aufbau von Systemen vorliegt. Die Bausteine wiederum stellen in sich geschlossene Funktionsblöcke dar, die durch Kombination mit anderen Bausteinen und Komponenten zu kompletten Systemen komplettiert werden können. Gleichzeitig kann so die Wiederverwendbarkeit bereits entwickelter und getesteter Funktionsbibliotheken unterstützt werden.

Schnittstellen und Standards

Um die mechanische, elektronische und informationstechnische Kompatibilität der Funktionsgruppen sicherzustellen, ist die Definition von allgemeingültigen Schnittstellen erforderlich.

Technologie

Wie bereits beschrieben sind Entwicklungen im Bereich miniaturisierter Systeme zumeist stark technologieorientiert. Die Beschreibung der eingesetzten Technologie, der Fertigungsabläufe und der Aufbau- und Verbindungstechnik sowie Hinweise auf erfahrene Dienstleister und Fertigungspartner erleichtert den Zugang zur Technologie.

Layoutvorlagen

Unter Berücksichtigung der definierten Schnittstellen werden in üblichen Layoutprogrammen wie z.B. Eagle Vorlagen erstellt, die bereits schon wesentliche Anforderungen der Aufbau- und Verbindungstechnik sowie der geometrischen Anforderungen berücksichtigen und so den Entwicklungsablauf stark beschleunigen und gleichzeitig die Zuverlässigkeit erhöhen können.

Fertigungshilfsmittel

Werkzeuge, Ablaufbeschreibungen und Handlungsanleitungen können herstellerübergreifend genutzt werden. Voraussetzung hierfür sind die gemeinsamen Schnittstellen und die definierten Technologien.

Testwerkzeuge

Die definierten Schnittstellen der Hardware ermöglichen eine hohe Wiederverwendbarkeit von Testwerkzeugen, auch über eine Anwendung hinaus. Einmal geschaffene Werkzeuge können so herstellerübergreifend eingesetzt und weiterentwickelt werden.

Programmierwerkzeug

Das Programmierwerkzeug soll insbesondere der Anforderung Rechnung tragen, in diesem Bereich auf Expertenwissen verzichten zu können und durch grafische Unterstützung und in Verbindung mit vorliegenden Funktionsblöcken einfach eigene Programme erstellen und diese testen zu können.

Softwarebibliotheken

Die Nutzung fertig erstellter und getesteter Funktionsblöcke kann durch vorliegende Funktionsbibliotheken stark gefördert und unterstützt werden. Diese Funktionsblöcke können hardwareunabhängig oder hardwareabhängig sein und in Verbindung mit dem Programmierwerkzeug bereits frühzeitig in der Entwicklung eingesetzt werden und so das Hardware/Software-Codesign unterstützen.

4.3 Entwicklungsablauf

Um den unterschiedlichen Stufen der Entwicklung, vom Funktionsmuster bis zum Serienprodukt, Rechnung zu tragen, ist der Entwicklungsablauf der Entwicklungsmethodik wie in Abbildung 4.2 dargestellt in drei Stufen unterteilt [BAUER 2006].

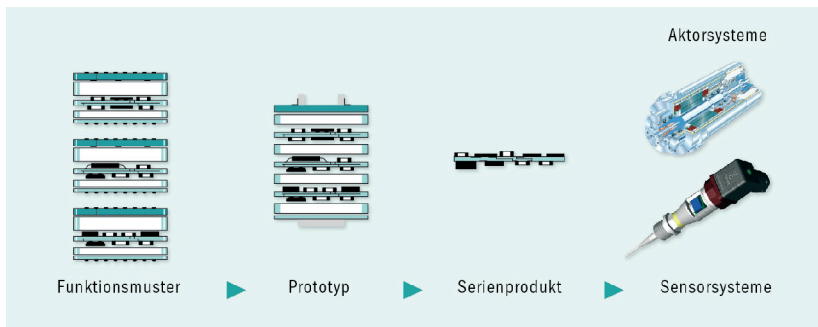


Abbildung 4.2: Entwicklungsablauf

Stufe 1: Funktionsmuster = Einsatz in der frühen Entwicklungs- und Testphase (entspricht A-Muster).

Vorkonfektionierte Funktionsgruppen werden über definierte Schnittstellen zu kompletten Systemen verbunden und können auch wieder getrennt werden. Parallel kann für den Softwareentwurf auf Bibliotheken zurückgegriffen werden. Dies ermöglicht neben dem schnellen Hardwareaufbau der Systeme auch einen parallelen Entwurfsprozess (Hardware/Software-Codesign) in der frühen Entwicklungsphase. Eine Wiederverwendung der vorkonfektionierten Baugruppen in verschiedenen Systemaufbauten und Anwendungen ist einfach möglich.

Stufe 2: Prototyp = Einsatz in der Entwicklung sowie zum Aufbau von Mustern und für Null- und Kleinserien (entspricht B-Muster).

Auf Basis der gleichen Funktionsgruppen von Stufe 1 werden Komplettsysteme erstellt, die bis hin zu Kleinserien eingesetzt werden können. Der Übergang von Stufe 1 zu Stufe 2 darf keinen zusätzlichen Entwicklungsaufwand im Bereich der Elektronik und Software erfordern, sondern sollte nur durch eine Modifikation im mechanischen Aufbau erfolgen.

Stufe 3: Anwendungsspezifisches mikromechatronisches System = Einsatz für Serienaufbau (entspricht C-Muster).

Unter Beibehaltung der Funktionsgruppen aus Stufe 1 und 2, aber Änderung des mechanischen Aufbaus, insbesondere aus Kostengesichtspunkten können Systeme mit gleicher Funktion erstellt werden. Hierfür werden die Schaltungen der vorliegenden Funktionsblöcke übernommen. Die erstellte Software ist sofort lauffähig.

Die Verwendung gleicher Funktionsblöcke in Hard- und Software in den unterschiedlichen Phasen der Entwicklung vereinfacht den Übergang vom Funktionsmuster zum Serienprodukt und erhöht gleichzeitig auch die Zuverlässigkeit der Systeme, da immer wieder auf getestete Teilfunktionen zurückgegriffen wird und gleichzeitig der Änderungsaufwand und somit auch der Testaufwand auf ein Minimum reduziert wird.

Aus diesem Drei-Stufen-Konzept resultiert eine wesentliche Entwicklungsaufgabe, die Definition eines Aufbaukonzepts, welches das Vorgehen nach dem Drei-Stufen-Konzept unterstützt. Parallel sind Werkzeuge zu definieren, die den Entwurfsablauf unterstützen.

Gleichzeitig kann durch die Entwicklungsmethodik EasyKit das Vorgehen nach VDI2206 und hier speziell das V-Modell unterstützt werden (siehe Abbildung 4.3). Funktionsbibliotheken und die damit verbundenen Möglichkeiten zum Hardware/Software-Codesign fördern den frühen Entwurfsprozess. Die begleitenden Werkzeuge für die Entwicklung, die Fertigung und den Test unterstützen domänenübergreifend den Entwurf. Das Drei-Stufen-Konzept wiederum führt zu einem einfacheren Übergang vom ersten Funktionsmuster zum Serienprodukt, die erforderlichen Anpassungen beim mehrfachen Durchlaufen des Makrozyklus können reduziert werden.

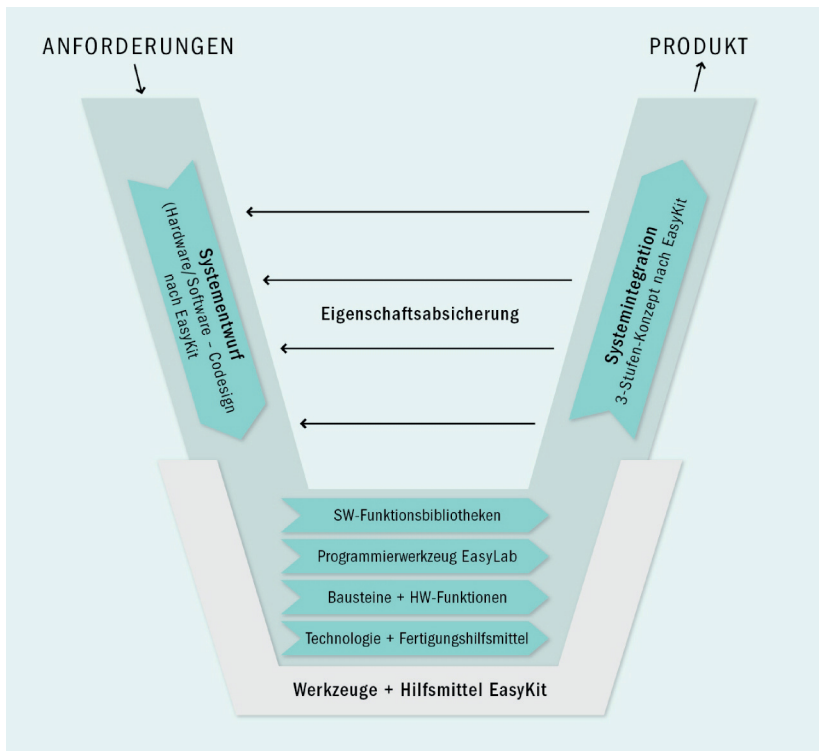


Abbildung 4.3: V-Modell unter Nutzung der EasyKit Entwicklungsmethodik

Dies führt gleichzeitig zu einem angepassten Entwurfsablauf (Abbildung 4.4), der die Möglichkeiten der Entwicklungsmethodik EasyKit berücksichtigt und bereits im Systementwurf das Hardware/Software-Codesign enthält. Wesentlicher Unterschied zum Entwurfsablauf des Baukastensystems Modulare Mikrosysteme (siehe Abbildung 3.4) ist dabei die Berücksichtigung der Softwareentwicklung, die im domänenspezifischen Entwurf parallel zu den Hardwareentwicklungsschritten stattfindet.

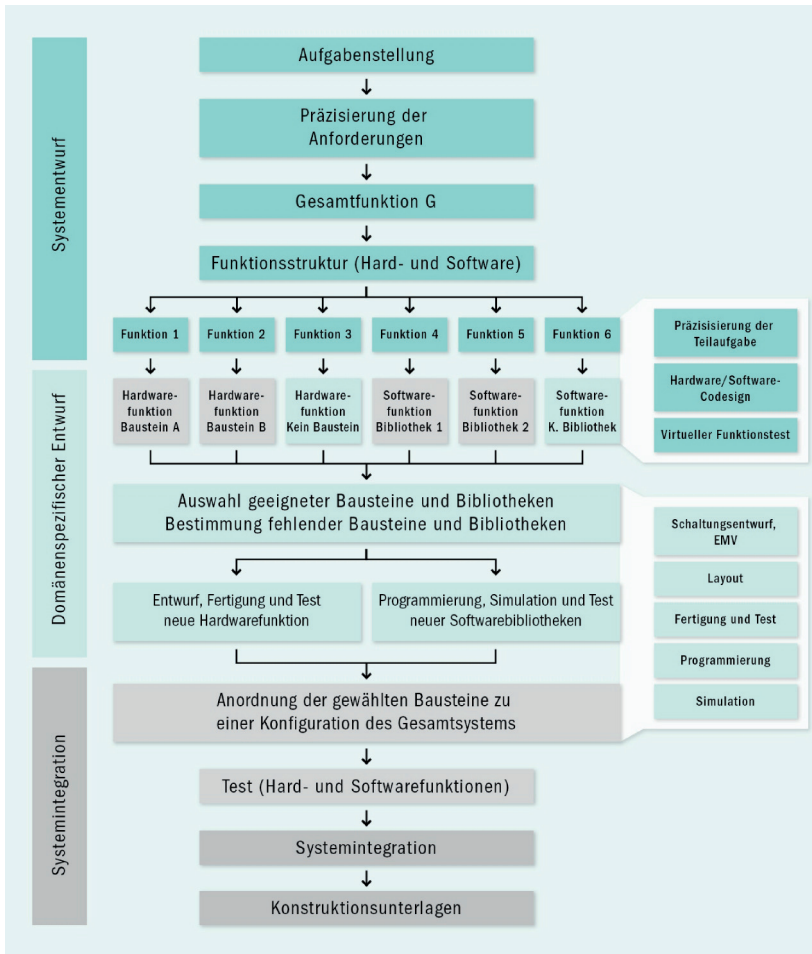


Abbildung 4.4: Angepasster Entwurfsalgorithmus für die Entwicklungsmethodik EasyKit unter Berücksichtigung des V-Modells

4.4 Entwicklungsaufgaben zur Umsetzung der Entwicklungsmethodik EasyKit

Aus der beschriebenen Struktur und dem Aufbau der Entwicklungsmethodik EasyKit resultieren die folgenden Anforderungen und der damit verbundene Entwicklungsbedarf:

- Entwurf einer Systemstruktur, die in den unterschiedlichen Entwicklungsstufen vom Funktionsmuster bis zur Serie möglichst viele wieder verwendbare Funktionen in Hardware und Software enthält.
- Die Handhabung muss für einen Anwender auf möglichst vielen Teilgebieten so gestaltet sein, dass ein Zugang nahezu ohne Expertenwissen in den Teildisziplinen erfolgen kann. Im Ergebnis kann ein mikro-mechatronisches System von einem einzelnen Entwickler oder einem kleinen Entwicklerteam domänenübergreifend eigenständig entwickelt und aufgebaut werden.
- Die Fertigung des Systems muss in normalen Produktionsumgebungen (z.B. Elektronikwerkstatt) ohne aufwendiges Produktionsequipment möglich sein.
- Das System muss eigenständig erweitert werden können und somit Firmen die Möglichkeit bieten, neben frei verfügbaren Funktionen auch eigene Funktionen in Hardware und Software so zu gestalten, dass firmeneigenes Know-how erhalten bleibt und für einen Dritten nicht offen gelegt wird.
- Das System ist als offenes System mit frei zugänglichen Schnittstellen zu gestalten. Durch die Entwicklungsmethodik können Entwicklungszeiten und Entwicklungskosten reduziert werden.
- Expertenwissen und Expertenwerkzeuge können für den Entwurfsprozess reduziert und in einigen Fällen vermieden werden. Hierfür ist der Entwurfsprozess einfach und verständlicher zu gestalten.
- Die Entwicklungsmethodik und die Werkzeuge sind geeignet für den Einsatz vom Funktionsmuster bis zur Serie.

5 Technologiekonzept für die Hardware der Entwicklungsmethodik EasyKit

Das Technologiekonzept umfasst die Aufbau- und Verbindungstechnik für die Bausteine und Systeme der Entwicklungsmethodik EasyKit. Dabei gelten die folgenden Forderungen:

- Für Stufe 1 liegen einzelne elektronische oder elektromechanische Funktionsgruppen in Form von eigenständigen Bausteinen vor. Diese verfügen über Schnittstellen, die eine einfache mechanische und elektrische Verbindung dieser Bausteine ermöglichen.
- Für Stufe 1 sind die Bausteine so zu gestalten, dass eine Wiederverwendung der Hardware gewährleistet werden kann.
- Der Aufwand für die Entwicklung und Fertigung von Bausteinen der Stufe 1 ist gering zu halten, so dass ein Aufbau in jeder Elektronikfertigung oder auch in einem Entwicklungslabor mit marktüblichen Werkzeugen realisierbar ist.
- Die Funktionsgruppen der Stufe 1 sind beim Übergang zu Stufe 2 hinsichtlich ihrer elektrischen oder elektromechanischen Basisfunktion unverändert zu lassen.
- Der Übergang von Stufe 1 zu Stufe 2 soll mit marktüblichen Fertigungswerkzeugen einfach realisierbar sein.

Unter Berücksichtigung dieser Anforderungen wurden unterschiedliche Aufbaukonzepte für Stufe 1 sowie Verbindungskonzepte für Stufe 1 und 2 der Entwicklungsmethodik konzipiert, aufgebaut und hinsichtlich ihrer Eignung bewertet.

Für das Baukastensystem Modulare Mikrosysteme wurde bereits ein 3D-Stacking-Ansatz [Leutenbauer 1997B] gewählt und die Funktionsgruppen in Form von Bausteinen aufgebaut. Die Bausteine können in unterschiedlichen Größen aufgebaut werden. In der so genannten Vorzugsgröße von 12,5mm x 12,5mm Grundfläche verfügen die Bausteine auf der Ober- und Unterseite über ein Grid-Array mit 63 Anschlüssen. Diese sind im Rastermaß von 1,27mm und einer Matrix von 8x8-1 angeordnet, wobei der fehlende Pin die Orientierung sicherstellt. Die eigentliche Funktionsgruppe ist im Inneren des Bausteins,

typischerweise auf den Rückseiten der Deckel- und Bodenelemente, angeordnet. Die Verbindung zwischen Deckel und Boden erfolgt über ein Rahmenelement. In diesem sind alle elektrischen Verbindungen durchgeführt. Diese Verbindungen sind als Vias im Inneren des Rahmens angeordnet.

Beim Aufbau von Systemen unter Industriebedingungen, basierend auf diesem technischen Ansatz kam es zu erheblichen Problemen. Diese waren im Wesentlichen:

- Die Testbarkeit konnte nicht gewährleistet werden. Nach dem Aufbau lagen alle Kontakte im Inneren und waren nicht zugänglich.
- Die innen liegenden Durchkontaktierungen verfügten über ein sehr kleines Rastermaß von 0,65mm. Die Verarbeitung in zwei oder mehr Ebenen führte zu vermehrten Kurzschlüssen bzw. offenen Verbindungsstellen, die wiederum aufgrund der Nicht-Zugänglichkeit nicht erkannt wurden.
- Die Verbindung zwischen den Bausteinen über Grid Arrays wurde bei der Verarbeitung im Nutzen schwierig. Durch das sehr hohe Gewicht der einzelnen Ebenen in Verbindung mit dem 3D-Aufbau trat der eigentliche Schwimmeffekt bei den Grid Arrays nicht mehr auf. Auch in diesem Bereich war eine erhöhte Anzahl von Kurzschlüssen zu festzustellen.

Aus diesen Gründen konnte sich die Aufbauform und damit verbunden das Baukastensystem Modulare Mikrosysteme am Markt nicht durchsetzen.

5.1 3D-Aufbautechnik für Bausteine der Stufe 1

Ziel von Stufe 1 des Entwicklungskonzepts ist der Aufbau von Funktionsmustern, indem jedoch bereits möglichst viele Funktionsgruppen eines späteren Systems Verwendung finden. Die Funktionsgruppen sollten in sich geschlossen sein und eine einfache Handhabung ermöglichen. Gleichzeitig muss es aber auch für einen Dritten einfach möglich sein, eigene Funktionsgruppen zu erstellen und in der vorgeschlagenen Aufbauform mit einfachen Mitteln der Laborfertigung zu realisieren. Für die Auslegung des Systems erfolgte eine Orientierung an den Vorzugsbausteinen des vorliegenden Baukastensystems Modulare Mikrosysteme, d.h. das System sollte in einem Rohr mit einem Innendurchmesser von 18mm integrierbar sein, bzw. bei quadratischer Auslegung eine Grundfläche von 12,5mm x 12,5mm aufweisen.

5.1.1 Aufbauform 1 – Funktionsträger mit Anschlüssen für steckbare Rahmen

Die erste Aufbauform orientiert sich stark an den vorliegenden Bausteinen und versucht insbesondere, den Nachteil der Verbindungstechnik zu beseitigen. Hierfür wurden die gelöteten Rahmen unter Beibehaltung der geometrischen Anordnung und der Anzahl der Kontaktstellen auf eine Variante mit Kunststoffrahmen umgestellt. Dieser Kunststoffrahmen enthält Stifte, die im vorgegebenen Rastermaß von 0,65mm angeordnet sind, um so die mechanische und elektrische Verbindung zu sichern.

Gleichzeitig wurde dem erhöhten Bedarf an Bestückungsfläche Rechnung getragen und ein Funktionsträger eingeführt, der beidseitig bestückbar ist. Aufbauform 1 besteht somit wie in Abbildung 5.1 dargestellt aus den Grundelementen:

- Funktionsträger
- Kunststoffrahmen mit Stiften
- Deckel- und Bodenplatte

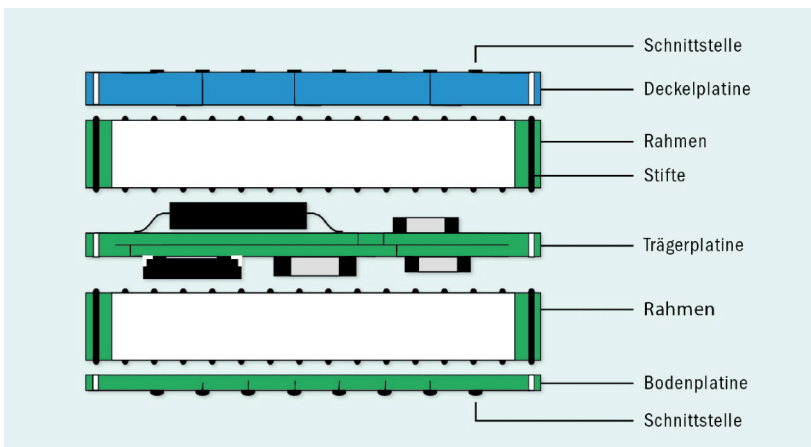


Abbildung 5.1: Aufbauform 1 – Funktionsträger für steckbare Rahmen

Zentrales Element dieser Aufbauform ist der Funktionsträger. Diese enthält alle Komponenten und stellt somit eine eigenständige Funktionsgruppe dar. Der Grundaufbau ist eine zweiseitig bestückbare Leiterplatte. Diese kann je nach

Bedarf zwei- oder mehrlagig ausgeführt sein. In den Grundabmessungen erfolgt eine Orientierung an der Vorzugsgröße der Bausteine des Baukastensystems Modulare Mikrosysteme mit 12,5mm x 12,5mm. Der Funktionsträger enthält die komplette Funktionsgruppe, so dass alle erforderlichen Verbindungen innerhalb dieser Funktionsgruppe direkt auf der Leiterplatte erfolgen können. Am Rand der Funktionsgruppe befinden sich umlaufend Vias mit einem Durchmesser von 0,35mm im vorgegebenen Rastermaß von 0,65mm.

Zweites Element ist der Rahmen. Dieser als Kunststoffspritzgussteil ausgeführte Träger mit den Kantenlängen von 12,5mm x 12,5mm verfügt über 64 Stifte, die ebenfalls im Raster von 0,65mm auf allen 4 Seiten umlaufend angeordnet sind (siehe Abbildung 5.2). Der Rahmen hat in der vorliegenden Ausführung eine Bauhöhe von 2,1mm. Die Breite des Rahmens beträgt 0,9mm, so dass im Inneren ein Bauraum von 10,7mm² zur Verfügung steht. Die quadratischen Stifte haben eine Kantenlänge von 0,25mm. Die Stifte ragen auf beiden Seiten 0,7mm aus dem Rahmen heraus und sind an beiden Enden angefasst, um ein einfacheres Einführen der Kontakte in die Hülsen der Leiterplatte zu ermöglichen. Die Kontaktstifte sind mit einer Ni/Au-Oberfläche ausgeführt.

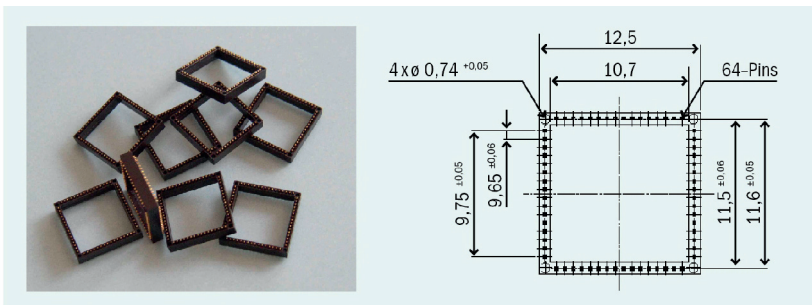


Abbildung 5.2: Aufbauform steckbarer Kunststoffrahmen

Abschließendes Element sind die Deckel- und die Bodenplatine. Diese verfügen über die Umverdrahtung des internen Bussystems auf das Grid Array. Hierfür verfügen diese Platinen wiederum am Rand über umlaufende Vias mit einem Durchmesser von 0,35mm.

Zur Verbindung von Rahmen und Leiterplatten stehen prinzipiell zwei Technologien zur Verfügung, die Einpresstechnik oder das Löten mittels Pin in Paste. Bei der

Einpresstechnik wird das Via der Leiterplatte so gefertigt, dass sich die Ecken der Stifte leicht in die Hülse eingraben/einpressen und somit einen festen Sitz gewährleisten. Da die Stifte auf ca. 0,2mm angefast sind, muss der Stift wenigstens 0,6mm tief eindringen, um einen sicheren Sitz zu gewährleisten. Die andere Alternative ist das Pin-in-Paste Löten. Hierfür sind die Vias mit einem Übermaß gefertigt, so dass die Stifte sicher und zuverlässig eingepasst werden können. Auf die Vias wird ein Lotreservoir gedruckt und der Rahmen in dieses eingesetzt. Beim manuellen Löten hingegen muss dieses Lot direkt zugeführt werden.

In einem ersten Schritt wurde die Einpresstechnik getestet. Für die weitere Fertigung ist ein Einpresswerkzeug (Kniehebelpresse) zu verwenden. In dieses kann das Rahmenelement eingeführt und in die entsprechende Platine eingepresst werden. Es empfiehlt sich, zuerst jeweils einen Rahmen in die Bodenplatte und die Deckelplatte zu pressen. Anschließend wird die Trägerplatte zwischen diesen beiden Platinen eingepresst. Der Funktionsträger muss somit eine Dicke $\geq 1,4\text{mm}$ haben.

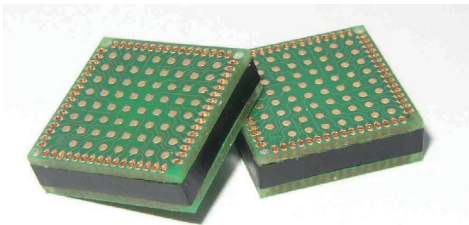


Abbildung 5.3: Bausteine mit Kunststoffrahmen

In der Praxis zeigte sich jedoch, dass aufgrund von Fertigungstoleranzen das Einpressen nicht zufriedenstellend verlief. Da die Stifte nur in den Rahmen eingepresst waren, wurden bei ungleichmäßigem Druck einzelne Stifte herausgeschoben. Diesem Effekt kann mit einer verbesserten Matrize entgegengewirkt werden. Diese wurde jedoch aufgrund von Aufwand und Kosten nicht mehr aufgebaut. In Summe konnte aber über alle Stifte eine gute und zuverlässige Verbindung realisiert werden. Bei weiterer Optimierung kann diese Technik somit eingesetzt werden.

Für die Pin in Paste Technologie ist sind Leiterplatten mit einem größeren Via-Durchmesser für die Stifte des Rahmens erforderlich. In diese wird das Lot eingedruckt. Voraussetzung hierfür ist das Vorliegen eines Siebes und das Rakeln

des Lots. Dies wird als entscheidender Nachteil gesehen, da dieses Equipment in vielen kleineren Labors nicht verfügbar ist. Alternativ kann auch jeder Kontakt einzeln gelötet werden. Der Aufwand hierfür ist jedoch sehr hoch. In Summe ist die Variante des Einpressens dem Löten vorzuziehen.

Nachteilig erwies sich in dieser Aufbauvariante noch immer die Testbarkeit. Die Kontaktstellen sind durch Sichtkontrolle nicht prüfbar. Bei der Einpressvariante sind jedoch aufgrund der Technologie Kurzschlüsse zwischen den Bahnen ausgeschlossen. Eine offene Verbindung hingegen, insbesondere im Inneren liegend, z.B. zwischen Funktionsträger und Rahmen kann nur durch einen elektrischen Funktionstest ermittelt werden. Darüber hinaus ist bei einem mehrstufigen Aufbau (mehrere Funktionsebenen übereinander) ein elektrischer Zugriff auf einzelnen interne Buskontakte überhaupt nicht mehr möglich.

Ein weiterer Nachteil dieser Bauform ist die Entwärmung. Durch den geschlossenen Aufbau, die hohe Packungsdichte beim Stapeln und die Verwendung thermisch fast ausschließlich schlecht leitender Materialien kann die im Inneren der Bausteine erzeugte Verlustleistung nicht zufrieden stellend nach Außen geführt werden. Im Bausteinstapel sind thermische Verlustleistungen von 0,1W pro Baustein für alle Materialien unkritisch. Für darüber hinausgehende thermische Verlustleistungen erfolgte eine modellbasierte thermische Abschätzung (siehe Abbildung 5.7, linke Abbildung), wobei sich diese Aufbauform als sehr ungünstig erwies.

5.1.2 Aufbauform 2 – Funktionsträger mit außenliegenden, lötbaren Anschlusskontakten

Die zweite Aufbauform wurde insbesondere unter der Forderung einer besseren Testbarkeit konzipiert. Während der Grundaufbau mit einem Funktionsträger, den Rahmen sowie Deckel und Boden identisch Aufbauform 1 ist, wurde die Geometrie und Anordnung der Kontakte des inneren Bussystems so geändert, dass diese von außen einfach zugänglich sind. Die Aufbauform 2 besteht somit aus:

- Funktionsträger
- Rahmenelement auf FR4-Basis
- Deckel- und Bodenplatte

Der grundlegende Aufbau ist wie beschrieben analog Aufbauform 1. Der Funktionsträger kann als zweiseitig bestückbare Leiterplatte ausgeführt werden.

Grundlegend für diese Variante ist die Konzeption eines neuartigen internen Bussystems. Dieses basiert auf dem Ansatz, den internen Bus an den Außenkanten des Bausteins und somit offen zu führen. In einem ersten Schritt wurde ein fertigungstechnisches Konzept für diese Bausteine erarbeitet. Um die Kurzschlussgefahr zu minimieren wurde der interne Bus für einen Baustein der Größe 12,5mm x 12,5mm auf 52 Signale reduziert. Dies bedeutet, auf jeder Seite des Bausteins sind 13 Signale mit einem Abstand von 0,8mm zu führen. Des Weiteren wurden die Möglichkeiten für die Erzeugung eines internen Busses an den Außenkanten der Bausteine überprüft. Da für die Fertigung auch weiterhin auf die etablierte Leiterplattentechnologie zurückgegriffen werden soll, wurde ein Aufbauschema konzipiert, bei dem die Lage der Durchkontaktierungen so verschoben wurde, dass durch einen entsprechenden Trennprozess eine offen liegende Kontaktfläche entsteht.

Sowohl die elektrische als auch die mechanische Verbindung werden über diese Kontakte realisiert. Die Vias haben einen Durchmesser von 350µm mit einem Restring von 100µm auf jeder Seite. Des Weiteren schließt sich direkt an den Restring auf beiden Seiten eine Kontaktfläche von 550µm x 700µm an. Der Lötstopplack wird so strukturiert, dass er eine Barriere zur Vermeidung von Kurzschlüssen darstellt (ursprüngliche Funktion) und gleichzeitig einen Abstandshalter, der zwischen zwei Komponenten einen definierten Abstand von zweimal Lötstopplackhöhe, d.h. ca. 70µm realisiert. Somit bildet sich eine Kavität zwischen zwei gegenüberliegenden Kontaktflächen, in die beim Lötprozess das Lot fließen kann. Dieses Lot ist die wesentliche Komponente für eine stabile und zuverlässige mechanische Verbindung zwischen zwei Komponenten.

Um die Montage der Funktionsträger zu erleichtern, wurden zusätzlich drei Hilfsbohrungen mit einem Durchmesser von 2 x 0,8mm und 1 x 0,7mm vorgesehen. Über Stifte können die einzelnen Komponenten so einfach zueinander ausgerichtet werden (siehe auch Abbildung 5.4). Dabei ist die Grundforderung, dass die Komponenten mit einer Genauigkeit von $\pm 0,1\text{mm}$ zueinander ausgerichtet werden. Bei dieser Forderung sind die Toleranzen im Herstellungsprozess zu berücksichtigen. Die drei Bohrungen können mit üblichen Fräsbohrplottern jeweils mit einer absoluten Genauigkeit von 0,1mm realisiert werden, wobei die 3 Bohrungen zueinander noch einmal einen relativen Fehler von 0,05mm aufweisen können. Bei Verwendung von Stiften mit einem Untermaß von 0,1mm kann stets die geforderte Toleranz eingehalten werden.

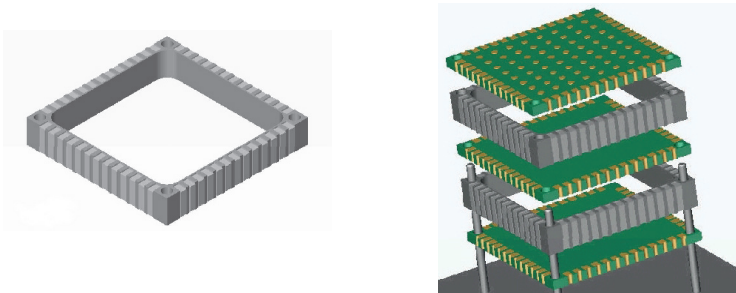


Abbildung 5.4: Konzept mit außen liegendem Bus

Der eigentliche Lötprozess kann im einfachsten Fall durch einen konventionellen, manuellen LötKolben mit einer Meißelspitze (z.B. mit einer Spitze von 0,6mm) durchgeführt werden. Aufgrund des gewählten Abstandes zwischen den Kontaktflächen treten Kurzschlüsse bei der Verarbeitung mit der richtigen Lotmenge nicht auf. Bei Verwendung von zu viel Lot können bei Bedarf Verbindungen zwischen den Bahnen gezielt hergestellt werden. Aufgrund der offen liegenden Kontakte sind die Verbindungsstellen per Sichtkontrolle unter dem Mikroskop einfach prüfbar.

Die Breite des Rahmens konnte in Versuchen bis auf 0,9mm minimiert werden, so dass im Inneren ein Bauraum von 10,7mm x 10,7mm (ca. 228mm²) zur Verfügung steht. Die Höhe des Rahmens kann variabel gestaltet werden. Für die Versuche wurden 3 Rahmenhöhen (1,2mm; 2,2mm und 3,2mm) gewählt, die mit Standardprozessen der Leiterplattenfertigung hergestellt wurden.

Die gesamte Fertigung der einzelnen Elemente (Funktionsträger, Rahmen, Deckel und Boden) erfolgt jeweils im Nutzen. Hierfür wurde eine Basisanordnung mit 5x5 Elementen im Nutzen gewählt (Abbildung 5.5). Durch entsprechende Anordnung und Vorgaben an den Leiterplattenhersteller können die gefertigten Elemente ohne Nachbearbeitung direkt weiter verarbeitet werden. Dabei besteht die Möglichkeit, im Rahmen der Bestückung des Funktionsträgers auch gleich die Rahmenelemente mit maschinell zu bestücken und somit den Fertigungsprozess wirtschaftlich zu optimieren.

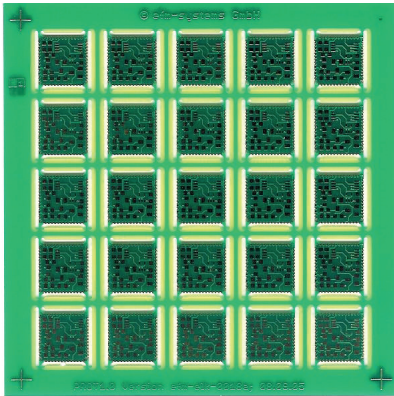


Abbildung 5.5: Nutzen mit Funktionsträgern

Im industriellen Umfeld können die Bausteine teilautomatisiert mittels eines selektiven Lötprozesses gefertigt werden. Positiv anzumerken bei dieser Aufbauform ist neben der Möglichkeit der Sichtkontrolle ebenfalls der einfache Zugriff auf das interne Bussystem und somit eine gute Testbarkeit. Die außen liegenden Kontakte können über Stifte einfach kontaktiert werden. Somit ist neben dem Kurzschluss-test von benachbarten Anschlüssen auch ein Multifunktions-test möglich.

Für Versuche und kleinere Serien erwies sich das manuelle Löten als effektiv. Dabei wird jede einzelne Lötbahn des Bausteins nacheinander verlötet. Anschließend können die Lötbahnen per Sichtkontrolle auf Kurzschlüsse und saubere Lotkontakte überprüft werden. Die Zuverlässigkeit in der Fertigung und die Möglichkeiten der Kontrolle bzw. des Test sind sehr zufrieden stellend.



Abbildung 5.6: Vom Funktionsträger (links) über ein Modul mit Rahmenelementen (Mitte) zum kompletten Baustein (rechts)

Darüber hinaus erwies sich diese Aufbauform als sehr vorteilhaft in der Entwärmung. Der Funktionsträger sollte hierzu zumindest 4-lagig mit ausreichend Kupfer in den Innenlagen ausgelegt sein. Durch die Anbindung an die außenliegenden Kontakte und die damit verbundenen guten thermischen Transportwege und die große metallische Außenfläche findet ein sehr guter Wärmetransport statt, auch größere Verlustleitungen können so nach außen geführt werden. Dies wurde durch Untersuchungen der Firma Temperaturmesstechnik Geraberg GmbH sowie Simulationen der Firma AMIC GmbH bestätigt [LEUTENBAUER 1997A , GROßER 2003]. In Abbildung 5.7 ist bei identisch angenommener Verlustleistung links ein Baustein-stapel mit extrem schlechtem thermischen Aufbau (jeweils 2-lagig, geschlossener Aufbau ohne metallische Außenflächen) und rechts ein Aufbau mit mehrlagigen Innenflächen und sehr guten Wärmetransport über die außen liegenden Kontakte. Somit können die HotSpot-Temperaturen im Inneren deutlich reduziert werden.

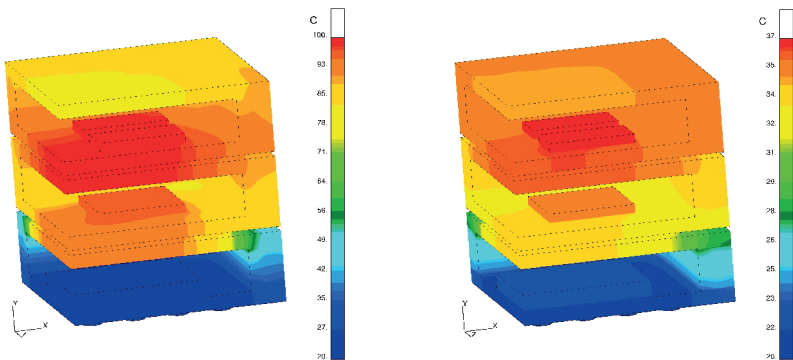


Abbildung 5.7: Thermische Simulation zur Wärmeverteilung in Bausteinen (Quelle: Amic GmbH)

5.1.3 Aufbauform 3 – Funktionsträger in Starr/Flex-Technologie

Unter Verwendung der Starr/Flex-Technologie [AMIRI JAM 2004] wurde in Kooperation mit der Firma digiraster Tetzner GmbH eine weitere Aufbauform entwickelt, bei der eine optimierte Ausnutzung des Bauraums eines Funktionsträgers sowie eine effiziente Fertigung im Mittelpunkt standen. Sowohl bei Aufbauform 1 als auch 2 stehen von der Grundfläche von 12,5mm x 12,5mm nur max. 10,7mm x 10,7mm zur Verfügung, d.h. bei zweiseitiger Bestückung ca. 228mm². Insbesondere bei CPU-Bausteinen ist die komplette Funktionalität nur mit Prozessoren sehr kleiner Bauform (6 x 6mm²) realisierbar.

Zur Vermeidung dieser Nachteile wurden in Aufbauform 3 zwei wesentliche Änderungen eingeführt. Zuerst wurde eine runde Bauform gewählt, um den Bauraum besser auszunutzen. Darüber hinaus wurde der interne Bus über einen flexiblen Schaltungsträger realisiert. Dies führt auch zu einer Änderung im mechanischen Konzept des Aufbaus. Da über den flexiblen Schaltungsträger die mechanische Stabilität nicht gewährleistet werden kann, wurden in einer ersten Version Stifte verwendet.

Die Aufbauform besteht aus zwei Platinen mit einem Durchmesser von 17,9mm, die über einen flexiblen Schaltungsträger miteinander verbunden sind (Abbildung 5.8). Auf der Oberseite der Platine ist das im VDMA Einheitsblatt 66305 definierte Grid-Array angeordnet, auf der Unterseite steht der Bauraum für die Funktion des Bauraums zur Verfügung. Unter Berücksichtigung eines Verlusts an Bauraum an der Anschlussseite des flexiblen Schaltungsträgers und eines Abstand zum Rand von 0,5mm stehen ca. 400mm² zur Verfügung (Vergleich: Bauform 2 = 228mm²).

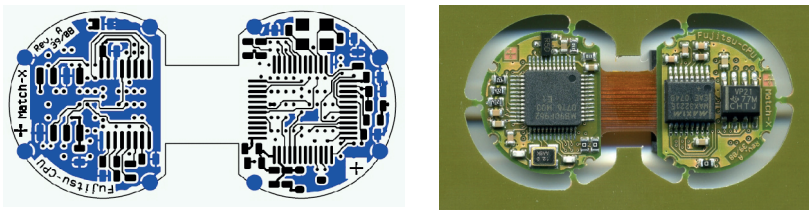


Abbildung 5.8: Links: Layout eines Funktionsträgers in Starr-Flex-Technologie, rechts: Bestückter Funktionsträger vor dem Falten

Die mechanische Verbindung der beiden Platinen erfolgt über 4 Stifte, die gleichzeitig für elektrische Funktionen (Masse- und Leistungsverbinding) wie auch für die Abführung von thermischer Energie genutzt werden können. Bei der Verarbeitung werden die beiden Funktionsträger zusammengeklappt, die Stifte eingeführt und verlötet. Bei Bedarf kann zwischen diese Lagen eine Abschirmplatte (ebenfalls mit einer Funktion für die thermische und elektrische Entkopp- lung) eingebracht werden.

In einer zweiten Aufbauvariante wurde auf die Stifte verzichtet und stattdessen ein ausgestanzter Distanzhalter aus Kunststoff eingesetzt. In Verbindung mit dem Gehäuse (Abbildung 5.9) konnte so eine sehr gute Stabilität des Aufbaus gewährleistet werden.

Die Bestückung des Funktionsträgers ist identisch den Bauformen 1 und 2. Da alle Funktionskomponenten des Bausteins auf der gleichen Seite bestückt werden, ist in diesem Zustand auch ein einfacher Test möglich. Nach dem Zusammenklappen des Bausteins erfolgt nur noch die Verbindung der Stifte, d.h. die Zuverlässigkeit und Testbarkeit des Fertigungsprozesses ist hoch. Bei einer Fortsetzung des Verbindungsprozesses in die dritte Dimension über das Grid Array sind aber die Zugriffsmöglichkeiten für den Test begrenzt, da alle Kontakt im Inneren des Funktionsträgers liegen bzw. über den flexiblen Schaltungsträger geführt sind. Eine Sichtkontrolle in diesem Zustand ist nicht möglich. Die Schritte von der Leiterplatte über den bestückten und gefalteten Baustein bis zum komplett gehäusten Baustein sind in Abbildung 5.9 dargestellt.

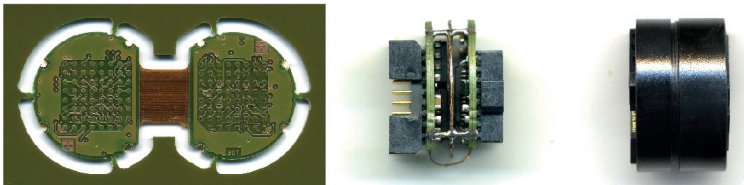


Abbildung 5.9: Links: Funktionsträger, Mitte: Bestückter und gefalteter Baustein, rechts: Kompletter Baustein im Gehäuse

Hinsichtlich der thermischen Eigenschaften des Aufbaus ist festzustellen, dass durch den prinzipiell offenen Aufbau eine gute Entwärmung stattfindet. Befindet sich der Baustein hingegen im Kunststoffgehäuse, ist vergleichbar Bauform 1 eine deutlich schlechtere Entwärmung festzustellen. Unter Verwendung eines gut thermisch leitenden Füllkörpers als Distanzhalter im Inneren kann hier aber eine gute Spreizung der thermischen Energie erreicht werden.

Weiterhin ist zu berücksichtigen, dass diese Aufbauform höhere Grundkosten (Einmal- und Fertigungskosten für die Leiterplatte) verursacht. Bei einer Fertigung in Stückzahlen größer 100 können diese Mehrkosten durch die bessere maschinelle Bestückbarkeit sowie eine einfachere Montage der Bausteine kompensiert werden und somit Vorteile gegenüber den anderen Aufbauformen erreicht werden.

5.1.4 Gegenüberstellung der Aufbauformen

Um die verschiedenen Aufbauformen miteinander vergleichen zu können erfolgt eine Gegenüberstellung und Bewertung nach verschiedenen Kriterien. Dabei werden alle aufgestellten Kriterien mit dem gleichen Faktor gewichtet. Es wird in 4 Stufen bewertet:

Sehr gut geeignet	Gut geeignet	Bedingt geeignet	Schlecht geeignet
4	3	2	1

Die erarbeiteten Aufbauformen werden nach folgenden Kriterien miteinander verglichen:

- Aufwand des Fertigungsprozesses: Für die Bewertung werden der erforderliche Arbeitsaufwand sowie das erforderliche Equipment zur Fertigung eines Bausteins berücksichtigt.
- Zuverlässigkeit des Fertigungsprozesses: Die Bewertung basiert auf der Nachvollziehbarkeit, Sicherheit und der Erreichbarkeit einer hohen Ausbeute bei der Fertigung.
- Möglichkeiten der Laborfertigung: Die Bausteine sollen auch in einem Elektroniklabor manuell fertigbar sein.
- Bauraum: Angestrebt wird eine möglichst optimale Nutzung des vorgegebenen Gesamtbauraums.
- Zugang zum elektrischen Funktionstest: Relevant für die Bewertung sind die Möglichkeiten des direkten Zugriffs auf den Funktionsträger sowie auf den internen Bus.
- Möglichkeiten der Sichtkontrolle: Relevant für die Bewertung sind die Zugangsmöglichkeiten für die Sichtkontrolle der Kontakt- und Verbindungsstellen.
- Möglichkeiten zu Nacharbeiten: Für die Bewertung werden vor allem die Möglichkeiten zur manuellen Nacharbeit, aber auch eines maschinellen Reworks berücksichtigt.
- Entwärmung: Relevant für die Bewertung ist die gute Abführung auftretender Verlustleistungen im Inneren des Bausteins.

- Kosten bei sehr kleinen Stückzahlen: Berücksichtigt werden die verhältnismäßigen Fertigungskosten für eine Stückzahl 1...10.
- Kosten bei mittleren Stückzahlen: Berücksichtigt werden die verhältnismäßigen Fertigungskosten für eine Stückzahl 50...500.
- Kosten bei großen Stückzahlen: Berücksichtigt werden die verhältnismäßigen Fertigungskosten für eine Stückzahl >1.000.

Kriterium	Aufbauform Modulare Mikrosysteme	Aufbauform 1	Aufbauform 2	Aufbauform 3
Aufwand des Fertigungsprozesses	2	1	3	3
Zuverlässigkeit Fertigungsprozess	1	3	4	4
Möglichkeit der Laborfertigung	1	2	4	1
Bauraum	2	2	2	4
Zugang elektrischer Funktionstest	1	1	4	3
Möglichkeiten der Sichtkontrolle	1	1	3	4
Möglichkeit zu Nacharbeiten	1	1	3	3
Entwärmung	1	1	4	3
Kosten bei kleinen Stückzahlen	2	1	4	1
Kosten bei mittleren Stückzahlen	2	1	3	3
Kosten bei großen Stückzahlen	2	3	2	4
	16	17	36	33

Tabelle 5.1: Gegenüberstellung der Aufbauformen

Aus den Ergebnissen der Bewertung in Tabelle 5.1 wird deutlich, dass die Aufbauformen 2 und 3 besonders geeignet sind. Die Aufbauform 3 ist insbesondere bei mittleren und großen Stückzahlen, bei denen auch die Laborfertigung nur eine untergeordnete Rolle spielt, die zu bevorzugende Aufbauform. Demgegenüber bietet die Aufbauform 2 insbesondere bei kleinen Stückzahlen die besten Möglichkeiten.

Prinzipiell sollte jedoch das Entwicklungskonzept dem Hersteller der Funktions-träger keine Einschränkungen bei der Fertigungstechnologie auferlegen, sondern nur Empfehlungen geben.

5.2 Verbindungstechnik Stufe 1

In Stufe 1 des Entwicklungskonzepts steht das Rapid Engineering bzw. Rapid Prototyping und damit der einfache und schnelle Aufbau von kompletten Systemen für den Einsatz in der frühen Entwicklungs- und Testphase im Fokus. Das in Kapitel 4 beschriebene Entwicklungskonzept geht von einer einfachen Verwendung und der Kombination definierter Funktionsblöcke aus. Diese sollten sich unkompliziert und schnell verbinden und optimalerweise auch wieder trennen lassen, so dass eine Wiederverwendbarkeit gewährleistet werden kann. Des Weiteren wird durch diese Aufbauform das Hardware/Software-Codesign unterstützt, d.h. im frühen Entwicklungsprozess kann sowohl auf definierte Hardware- als auch Softwarefunktionsblöcke zurückgegriffen werden. Eine optimale Auswahl der Funktionsblöcke unter Berücksichtigung der realen Ressourcen wird somit einfach unterstützt.

Als Ausgangspunkt für den Funktionsblock werden die in Kapitel 5.1.1 beschriebenen Bausteine definiert. Die Schnittstelle zwischen den Bausteinen sind somit die Grid-Arrays auf der Ober- und Unterseite mit einem Anschlussraster von 1,27mm und 8x8-1 Pins nach Vorgabe des VDMA-Einheitsblatts 66305. Die Ausführung dieser Grid-Arrays kann nach den Anforderungen der hier zu definierenden Verbindungstechnologie ausgelegt werden. Somit ist es freigestellt, ob diese Verbindungsstellen vergoldet, verzinnt oder in einer anderen Technologie ausgeführt sind.

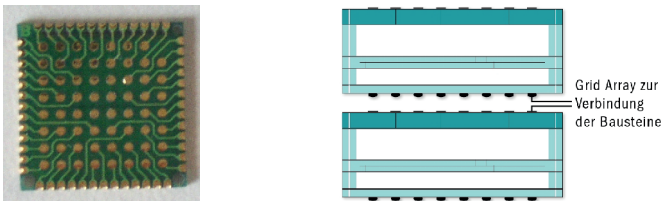


Abbildung 5.10: Links – Grid-Array als Schnittstelle zwischen den Bausteinen, rechts – verbindende Bausteine

Die Verbindung zwischen den Bausteinen ist so zu gestalten, dass sie technisch sehr zuverlässig sowohl die mechanische als auch die elektrische Verbindung der Bausteine sicherstellt. Auf Basis dieser Anforderungen wurden drei Verbindungsarten konzipiert und auf ihre Einsatztauglichkeit praktisch untersucht.

5.2.1 Verbindung über Federmatrizen

Für die erste Aufbauvariante wurden federnde Kontaktstifte als Lösungsvorschlag gewählt. Diese Kontaktstifte sind im Rastermaß des Grid-Array anzuordnen. Ein entsprechendes Bauelement ist am Markt nicht bekannt. Verfügbare einzelne Federstifte weisen eine vergleichsweise lange Bauform auf, die beim Stapeln mehrerer Bausteine zu extrem großen Bausteinstapelhöhen ($>100\text{mm}$) führen würde. Daher wurde gemeinsam mit einem japanischen Hersteller für Kontaktstifte (Shin Etsu) eine neue Aufbauform von Stiften gewählt. Dabei besteht der Kontaktstift aus 3 einzelnen Elementen (2 Stifte, 1 Feder) die im zusammengesetzten Zustand lediglich eine Bauhöhe von 2,7mm aufweisen. Für die Bausteine wurde durch den Hersteller ein Kunststoffgehäuse konstruiert, das den Einsatz von 63 Kontaktstiften im Raster von 1,27mm ermöglicht. Im zusammengebauten Zustand ermöglicht diese im Weiteren als Federmatrix bezeichnete Komponente das einzelne, unabhängige Betätigen aller Kontaktstifte mit einem Federweg von beidseits 0,3mm. Die Kontaktelemente bestehen aus Kupfer-Beryllium und sind mit einer galvanischen Goldschicht überzogen. Die Stifte sind vorgespannt und benötigen eine Kraft von 0,86N/Pin für den gesamten Weg.

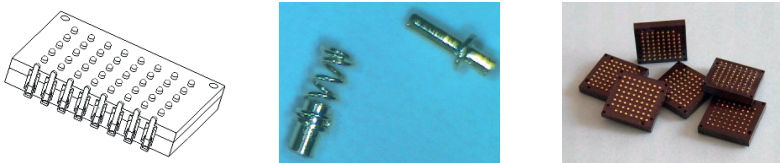


Abbildung 5.11: Elemente der Federmatrix

Über diesen Aufbau kann jedoch lediglich die elektrische Verbindung zwischen Bausteinen sichergestellt werden. Für die mechanische Verbindung wurde eine Spannvorrichtung konstruiert. Diese besteht aus einem Tubus sowie einer Spannschraube. Der Tubus hat eine Bauhöhe von 47 mm, verfügt im Inneren über einen quadratischen Freiraum mit einem Kantenmaß von 12,6mm und ist an zwei Seiten ausgespart, so dass ein einfaches, manuelles Einführen der Bausteine und der Federmatrizen ermöglicht wird.

In diesen Aufbau kann jeweils abwechselnd ein Baustein und eine Federmatrix eingeführt werden. Der erforderliche Druck zur sicheren Kontaktierung wird über die Spannschraube ausgeübt. Diese verfügt über ein Steilgewinde, die über zwei

Stifte im Tubus aufgenommen werden. Somit kann mit wenigen Umdrehungen ein großer Höhenunterschied realisiert werden. Dies ist erforderlich, um auch bei unterschiedlichen Baustein- und Systemhöhen eine schnelle Fixierung zu ermöglichen.



Abbildung 5.12: Tubus mit Bausteinen und Federmatrizen

Dieser Aufbau wurde diversen Praxistests unterzogen. Als positiv wurden die hohe Zuverlässigkeit der Federmatrix und die einfache Handhabung des gesamten Aufbaus herausgestellt. Negativ bewertet wurde eine unzuverlässige Fixierung des gesamten Aufbaus, die abnehmende Zuverlässigkeit der Verbindung bei mehreren Bausteinverbindungen innerhalb eines Stapels und das sehr individuell positiv wie auch negativ empfundene Einführen der Bausteine. Ein weiterer Nachteil ist, dass die Federmatrix vergoldete Oberflächen erfordert. Für einen häufigen Gebrauch müssen diese galvanisch vergoldet sein. Verzinnte Oberflächen sind problematisch, da die oxidierte Oberfläche von den hier verwendeten Kontaktstiften nicht durchdrungen werden kann. Hier sind jedoch mit anderen Stiftformen (z.B. Kronenstift oder spitze Stiftform) bessere Ergebnisse zu erwarten. Aufgrund der Nichtverfügbarkeit dieser Stiftformen für die Federmatrix wurde dies nicht weiter untersucht.

Beim Aufbau von kompletten Systemen ist darüber hinaus die Lageorientierung der Bausteine wichtig. Da bei einem geschlossenen Baustein auch die Funktion nicht mehr optisch erkennbar ist, wurde ein Farbcode auf die Bausteine eingebracht (siehe Abbildung 5.12). Hierfür wurde eine Seite definiert und eine Code-tafel angelegt. Somit kann der Farbcode neben der Erkennung der Funktion auch zur Lageerkennung genutzt werden.

Die Federmatrix selbst ist ebenfalls lageabhängig, verfügt aber für die Orientierung die Montagehilfsbohrungen in den identischen Dimensionen wie die Bausteine. Da diese in der Spannvorrichtung nicht genutzt werden, ist die richtige Lageorientierung beim Stapeln nur über die auf den Bausteinen vorhandene Farbmarkierung

feststellbar und muss bei jedem Aufbau kontrolliert werden. Des Weiteren musste festgestellt werden, dass die Schnelligkeit und Zuverlässigkeit des Aufbaus stark vom Anwender und dessen individuellen Vorgehen abhängig waren.

Die Einkaufskosten für eine Federmatrix liegen bei 45,- EUR (ab 500 Stück) und sind vergleichbar mit den durchschnittlichen Herstellungskosten für einen Baustein. Darüber hinaus sind bei dieser Verbindung aber auch die Kosten für die Spannvorrichtung zu berücksichtigen. D.h., für einen typischen Systemaufbau mit 4 Bausteinen sind drei Federmatrizen sowie die Spannvorrichtung erforderlich, d.h. in Summe Kosten von ca. 400,- EUR (ohne Berücksichtigung der Bausteinkosten).

Die Anzahl der Kontaktzyklen wird vom Hersteller mit 1.000 angegeben. In der Praxis waren bei den verwendeten Versuchsaufbauten auch bei einer hohen Anzahl von Kontaktzyklen (>1.000) keine Ausfälle festzustellen. Weiterführende Untersuchungen hierzu wurden jedoch nicht durchgeführt.

5.2.2 Verbindung über Kontaktfolien

Für die zweite Aufbauvariante wurde eine Kontaktfolie ausgewählt. Diese anisotrop leitfähige Kontaktfolie besteht aus Silikonkautschuk und ist durchsetzt mit vergoldeten Drähten aus Kupfer-Beryllium, die einen Durchmesser von 40µm aufweisen. Die Abstände der Drähte können in den StandardbaufORMen 0,05mm, 0,1mm und 0,25mm aufweisen. Für die Kontaktierung der Bausteine weisen die Drähte ein Raster von 0,1mm x 0,25mm auf. Der Kontakt selbst entsteht durch Kompression der Folien.

Die Kontaktfolie hat in der vorliegenden Bauform eine Höhe von 1 mm und wird vom Hersteller in Platten mit Abmessungen von 50mm x 50mm geliefert, die beliebig zugeschnitten werden können. Die Drähte ragen auf beiden Seiten um ca. 40µm aus der Folie. Dieses Hervorstehen der Drähte ist zwingend erforderlich, um die Höhendifferenz des Lötstopplacks zur eigentlichen Kontaktfläche zu überwinden.

Bei einem Kontaktpad des Bausteins kommen je nach Lage der Folie 3-5 Drähte mit jedem Pad in Kontakt. Durch einen definierten Anpressdruck entsteht ein zuverlässiger elektrischer Kontakt zwischen Pad und Draht. Für eine optimale Verbindung sollte die Kompression der Folie bei 15...25% liegen. Eine Verbindung erfolgt sowohl bei vergoldeten Oberflächen als auch bei verzinnnten Oberflächen in gleicher Qualität. Bei verzinnnten Oberflächen durchdringt der dünne

Draht die Oxidschicht an der Oberfläche. Entsprechende Eindrücke sind per Sichtkontrolle unterm Mikroskop einfach zu visualisieren (Abbildung 5.13).



Abbildung 5.13: Kontaktfolie (12,5mm x 12,5mm) nach Benutzung mit sichtbaren Spuren der BGA

Für erste Versuche wurde ein Testadapter konstruiert, bei dem die Kontakte für einen einfachen Abgriff auf einen umlaufenden Stecker geführt wurden. Über einen Spannmechanismus, bestehend aus Führungsstiften und einer Druckplatte können die Bausteine gespannt werden, wobei durch das Konstruktionsprinzip ein paralleles Anpressen des Bausteins gewährleistet wird. Die exakte Ausrichtung des Bausteins erfolgte über die Montagestifte und die entsprechenden Bohrungen in den Bausteinen. Für den Aufbau kompletter Systeme wurde der bereits in Kapitel 5.2.1 beschriebene Spannadapter verwendet.

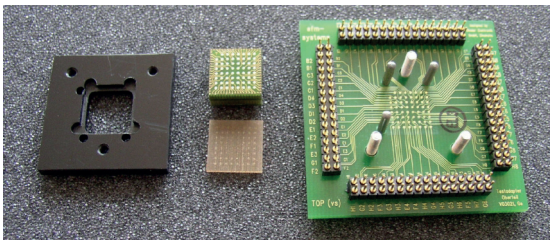


Abbildung 5.14: Testboard mit Kontaktfolie

Bereits in Verbindung mit dem Testboard konnte nur bei ca. 90% der Versuche eine Verbindung hergestellt werden. Als wesentliche Ursache hierfür konnte eine Beschädigung der Drähte ermittelt werden. Bei zu starkem Anpressdruck kommt es zu einem irreversiblen Abknicken der Drähte. Durch dieses Abknicken kann der Höhenunterschied zwischen Lötstopplack und Kontaktfläche beim folgenden Versuch nicht mehr überwunden werden. In weiteren Versuchen ließ sich nachweisen, dass bei einem Erstgebrauch der Kontaktfolie grundsätzlich eine Verbindung hergestellt werden konnte. Im Gegensatz hierzu stieg bei häufigerem

Gebrauch die Ausfallrate an. Eine bessere Kontrolle des Anpressdrucks ist nur mit größerem Aufwand möglich, in Verbindung mit den Fertigungstoleranzen des Bausteins letztendlich nur über ein sensorgesteuertes System. Dieser Aufwand wird für das Ziel, ein einfach bedienbares und für Nichtexperten nutzbares System zu entwickeln, als unangemessen gesehen. Daher wurde für weitere Versuche eine maximale Verwendbarkeit für 10 Zyklen definiert.

Wie bei der Federmatrix auch ist beim Aufbau von kompletten Systemen auf die Lageorientierung der Bausteine zu achten. Auch hier werden falsch eingesetzte Bausteine nicht erkannt und die richtige Lageorientierung ist nur über die auf den Bausteinen vorhandene Farbmarkierung feststellbar und muss bei jedem Aufbau kontrolliert werden. Für Versuche mit mehreren Bausteinen wurden jeweils abwechselnd ein Baustein und eine Kontaktfolie in das Spannsystem eingeführt. Die Herstellung der Verbindung erwies sich als sehr schwierig und war stark abhängig vom individuellen Vorgehen der Versuchsperson. In Summe konnte bei unter 50% der Versuche eine Verbindung hergestellt werden, wobei bei einzelnen Versuchspersonen überhaupt keine Verbindung zustande kam.

Die Kosten für eine Kontaktfolie liegen bei ca. 10,- EUR. Bei der Kostenbetrachtung ist jedoch vergleichbar der Federmatrix auch die erforderliche Spannvorrichtung zu berücksichtigen. D.h., für einen typischen Systemaufbau mit 4 Bausteinen sind drei Kontaktfolien sowie die Spannvorrichtung erforderlich, d.h. in Summe Kosten von ca. 280,- EUR. Des Weiteren ist zu berücksichtigen, dass bei einer Wiederverwendung nach max. 10 Zyklen neue Kontaktfolien eingesetzt werden müssen.

5.2.3 Verbindung über Matrixstecker

Als dritte Aufbauvariante wurde ein High-Density Steckverbinder (im Weiteren als Matrixstecker bezeichnet) ausgewählt, der im Raster von 1,27mm von der Firma Samtec [SAMTEC 2007] verfügbar ist. In einer ersten Version waren diese Matrixstecker nur in Rasteranordnungen von 8x20 Pins oder größer verfügbar. Somit mussten die Stecker mechanisch nachbearbeitet werden.

Für den praktischen Aufbau wurde ein Matrixstecker mit einer Rasteranordnung von 8x20 ausgewählt. Vorteilhaft an diesem Stecker sind die Außenabmessungen in der Breite (8 Pins) von 12,6mm, die auch dem Grundmaß des Funktionsträgers entsprechen. Der Stecker wird in der Länge auf ein Maß von 12,5mm abgefräst. Somit entsteht ein 8x8 Raster. Über das Ball Grid Array auf der Unterseite kann der Stecker leicht auf die vorhandenen Deckel- und Bodenplatten

aufgelötet werden. Der Matrixstecker selbst verfügt über eine Führung, die über das Kunststoffgehäuse realisiert ist. Diese Führung wird durch den Fräsprozess nicht beeinflusst. Des Weiteren verfügt der Matrixstecker über eine mechanische Orientierungshilfe. Auch diese wird durch den mechanischen Eingriff nicht beeinflusst. Die Bauhöhe eines Steckerpaares beträgt 5mm.

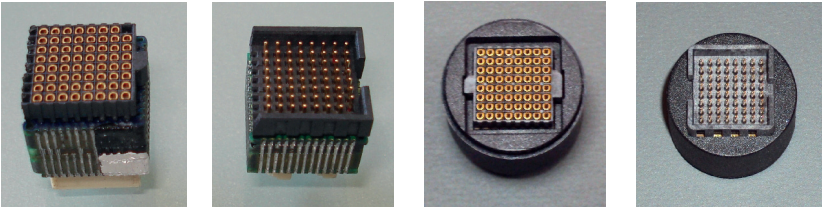


Abbildung 5.15: Matrixstecker im 8x8 Raster mit Orientierungshilfe auf Deckel- und Bodenplatte in Aufbauvariante 2 und 3

Die Kontaktelemente des Stiftsteckers sind aus Kupfer-Beryllium, vergoldet und weisen einen Durchmesser von 0,4mm auf. Die Kontaktelemente des Buchsensteckers sind ebenfalls vergoldete Stanz-Biegeteile aus Kupfer Beryllium, die in das Kunststoffgehäuse eingesetzt werden. Auf der Unterseite weisen beide Stecker ein Ball Grid Array auf. Die Stecker werden über einen Standardlötprozess, idealer Weise in der Dampfphase, auf die vorhandenen Deckel- und Bodenplatten der Bausteine (Aufbauvariante 2) bzw. auf die Außenseiten der Starr-Flex-Bausteine (Aufbauvariante 3) aufgelötet.

Für die Verbindung von Bausteinen sind bei Verwendung dieser Bausteine keine weiteren Hilfsmittel erforderlich (siehe Abbildung 5.16). Die Stecker realisieren sowohl die elektrische als auch die mechanische Verbindung der Bausteine. Für die Stecker werden 500 Steckzyklen garantiert.

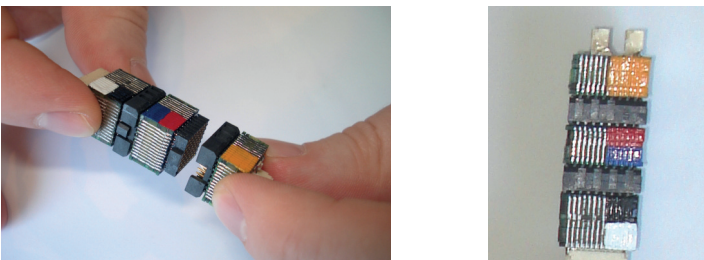


Abbildung 5.16: Bausteinstapel über Matrixstecker verbunden

Auch dieser Aufbau wurde unterschiedlichen Praxistests unterzogen. Sowohl die Qualität der elektrischen als auch der mechanischen Verbindung war sehr gut. Die Erfolgsquote bei der Inbetriebnahme von Komplettsystemen lag bei 100%. Ein individueller Einfluss des Bedieners auf das Ergebnis konnte demzufolge nicht festgestellt werden. Die mechanische Führung der Matrixstecker erwies sich ebenfalls als sehr zuverlässig. Positiv anzumerken ist, dass durch die mechanische Orientierung der Kunststoffgehäuse des Matrixsteckers ein fehlerhafter Aufbau der Bausteine ausgeschlossen werden kann. Jedoch sind die Pins des Stiftsteckers gegenüber mechanischen Einflüssen empfindlich und können bei unsachgemäßer Behandlung verbogen werden. Im Rahmen der Versuchsreihen kam es allerdings zu keinen derartigen Beschädigungen.

Durch die im Vergleich zu den anderen Aufbauvarianten höhere Bauform der Stecker werden höhere Baustein stapel erzeugt. Gleichzeitig sind die Abmessungen des Gesamtsystems durch den Wegfall der zusätzlichen mechanischen Verbindungselemente deutlich kompakter. Im Weiteren Verlauf der Entwicklung wurde gemeinsam mit dem Hersteller, der Firma Samtec, ein angepasster Stecker mit 8x8 Kontakten realisiert. Somit konnten der Aufwand für die Fertigung und die Kosten für die Beschaffung noch einmal deutlich reduziert werden. Die Beschaffungskosten für eine Verbindungsstelle (Stift- und Buchsenstecker) liegen bei Stückzahlen größer 500 bei ca. 10,-EUR pro Verbindungsstelle. Allerdings muss jeder Baustein mit einer Steckerpaarung ausgestattet werden, d.h. bei 4 Bausteinen liegen die Kosten bei ca. 40,-EUR.

5.2.4 Gegenüberstellung der Verbindungsarten

Um die verschiedenen Verbindungsarten vergleichen zu können, wurden wiederum unterschiedliche Kriterien ermittelt, die für den Einsatz relevant sind. Dabei werden alle aufgestellten Kriterien mit dem gleichen Faktor gewichtet. Es wird in 4 Stufen bewertet:

Sehr gut geeignet	Gut geeignet	Bedingt geeignet	Schlecht geeignet
4	3	2	1

Die erstellten Verbindungsarten werden nach folgenden Kriterien miteinander verglichen:

- Handhabung: Basis der Bewertung ist die Einfachheit der Handhabung, insbesondere unter Berücksichtigung der erstmaligen Anwendung des Systems.
- Verfügbarkeit: Zur Bewertung wird neben der Verfügbarkeit der eigentlichen Verbindungselemente auch die Verfügbarkeit der erforderlichen Hilfsmittel berücksichtigt.
- Zuverlässigkeit: Hierunter ist insbesondere die Zuverlässigkeit des eigentlichen Verbindungselements (Kontaktfedersystem, Kontaktdraht, Steckverbinder) zu verstehen.
- Baugröße (inkl. Hilfsmittel): Berücksichtigt werden die Baugröße des eigentlichen Verbindungselementes und die Baugröße der zwingend erforderlichen Hilfsmittel.
- Abhängigkeit von den individuellen Fähigkeiten des Anwenders: Für alle Verbindungselemente wurden Untersuchungen zur Anwendbarkeit in unterschiedlichen Zielgruppen durchgeführt. Die dabei festgestellten Abhängigkeiten von den individuellen, haptischen Fähigkeiten des Anwenders sind Grundlage dieser Bewertung.
- Anzahl der Verbindungszyklen: Bewertet wird die Anzahl der zulässigen Verbindungszyklen auf Basis der Herstellerangaben sowie der Erfahrungen bei den Tests der Verbindungselemente.

Federmatrix	Kontaktfolie	Matrixstecker
1.000	10	500

- Kosten: Bewertet werden die Kosten für den Aufbau eines Systems mit 4 Bausteinen unter Berücksichtigung der eigentlichen Verbindungselemente sowie der erforderlichen Hilfsmittel

Um eine Vergleichbarkeit der unterschiedlichen Varianten zu gewährleisten wurden folgende Annahmen getroffen:

- Ein System besteht aus 4 Bausteinen und somit 3 Verbindungsstellen zwischen den Bausteinen.

- In einem Baukasten befinden sich 10 verschiedene Bausteine, mit denen 25 verschiedene Systemkonfigurationen aufgebaut werden können.
- Das zusätzlich erforderliche Zubehör (Stecker, Anschlusskabel, Netzteil) wird für alle 3 Varianten als identisch angenommen, die Kosten betragen 100,00 EUR.

Daraus resultieren folgende Kosten:

	Federmatrix	Kontaktfolie	Matrixstecker
a) Durchschnittlicher Preis pro Baustein	75,00 €	75,00 €	90,00 €
b) Kosten zusätzlich erforderliche Kontaktelemente	150,00 €	30,00 €	0,00 €
c) Kosten zusätzlich erforderlicher Spannmittel	250,00 €	250,00 €	0,00 €
d) Preis Zubehör (Anschlussset)	100,00 €	100,00 €	100,00 €

Tabelle 5.2: Kostenpositionen für Berechnung der Systemkosten

Für den Kostenvergleich werden folgende drei Varianten miteinander verglichen:

1. Die Grundkosten (GK) für einen derartigen Baukasten.

$$GK = 10 \cdot a + b + c + d$$
2. Die Kosten für einen einzelnen Systemaufbau (SK).

$$SK = 4 \cdot a + b + c + d$$
3. Die anteiligen Kosten, wenn alle 25 Systemvarianten aufgebaut werden. Hierbei wird davon ausgegangen, dass in Aufbauvariante 2 für jede Systemvariante ein neuer Satz Kontaktfolien zum Einsatz kommt.

$$SV = GK/25 + VM$$

mit VM = Verbrauchsmaterial

Auf Basis dieser Annahmen ergeben sich die in Tabelle 5.3 aufgeführten Kosten:

	Federmatrix	Kontaktfolie	Matrixstecker
1. Kosten Basissatz mit 10 Bausteinen	1.250,00 €	1.130,00 €	1.000,00 €
2. Kosten für einzelnen Systemaufbau	800,00 €	680,00 €	460,00 €
3. Anteilige Kosten bei 25 Aufbauvarianten	50,00 €	74,00 €	40,00 €

Tabelle 5.3: Vergleich der Kosten der Aufbauvarianten

In allen 3 Fällen erweist sich der Matrixstecker als günstigste Aufbauform. Die Kontaktfolie hat bei einzelnen Systemaufbauten gegenüber dem Matrixstecker Vorteile, wenn jedoch eine höhere Variantenvielfalt und damit mehr Kontaktzyklen gefordert sind weist die Federmatrix gegenüber der Kontaktfolie deutliche Kostenvorteile auf.

Kriterium	Verbindung 1	Verbindung 2	Verbindung 3
Handhabung	3	2	4
Verfügbarkeit	2	3	3
Zuverlässigkeit	3	1	4
Baugröße (inkl. Hilfsmittel)	2	2	4
Abhängigkeit von den individuellen Fähigkeiten des Anwenders	3	1	4
Anzahl der Verbindungszyklen	4	1	3
Kosten	2	1	3
	19	11	25

Tabelle 5.4: Gegenüberstellung der Verbindungsarten

Im Ergebnis der Bewertung nach Tabelle 5.4 ist festzustellen, dass nur die Verbindungsform 3 in Form des Matrixsteckers den grundlegenden Anforderungen eines einfachen, schnellen und zuverlässigen Systems für das Rapid Prototyping genügt und darüber hinaus noch den größten Miniaturisierungsgrad aufweist. Die Federmatrix erwies sich am ehesten als Alternative zum Matrixstecker, wird aber aufgrund der erforderlichen Hilfsmittel als nur bedingt geeignet eingeschätzt.

5.3 Verbindungstechnik Stufe 2

Stufe 2 der Entwicklungsmethodik definiert den Aufbau von Prototypen sowie Null- und Kleinserien. Um diesen Aufbau effektiv und einfach zu gestalten, ist eine möglichst hohe Wiederverwendung von Funktionen und Baugruppen in Hard- und Software aus Stufe 1 anzustreben.

Der Ansatz zur Umsetzung dieser Forderung ist die Weiterverwendung der bereits in Stufe 1 der Entwicklungsmethodik definierten Funktionsträger. Die in Kapitel 5.1.4 ausgewählte Bauform bietet hierfür die grundlegenden Voraussetzungen. In Stufe 1 wird der Funktionsträger mit den Rahmen sowie den Boden- und Deckelementen mit den Matrixsteckern verbunden. Wenn der Funktionsträger nur mit Rahmenelementen bestückt wird, entsteht eine offene Baugruppe (Abbildung 5.17), die einfach mit üblichen Technologien der Leiterplattenfertigung weiter verarbeitet werden kann.

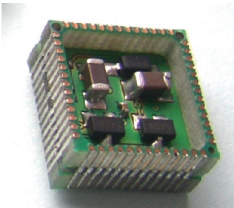


Abbildung 5.17: EasyKit Baugruppe bestehend aus Funktionsträger mit Rahmenelementen

Hierfür wird wiederum die Kontaktierungstechnologie mit den außen liegenden Kontakten verwendet. Die umlaufenden Kontakte im Raster von 0,8mm sind gut zugänglich. Dies betrifft sowohl eine manuelle Verarbeitung als auch eine Bestückung durch Automaten. Somit kann eine derartige Baugruppe vergleichbar einem normalen SMD-Bauteil gehandhabt und in planaren Schaltungen bestückt werden.

Ebenso ist die Weiterverarbeitung in Stapelbauweise einfach möglich. Bei Vorliegen mehrerer Funktionsträger mit Rahmenelementen können diese über die Montagebohrungen exakt zueinander orientiert werden. Das Löten kann wieder wie in Kapitel 5.1.2 beschrieben manuell oder automatisch erfolgen. Dies bedeutet, dass ein in Stufe 1 vorliegendes System bei Verwendung der identischen Funktionsträger durch Weglassen der Deckel- und Bodenelemente direkt in Stufe 2 überführt werden kann. Die funktionsbestimmte Hardware bleibt somit

unverändert. Dies führt gleichzeitig zu einer Reduzierung von Kontaktstellen im Gesamtaufbau eines Systems. Da die komplette Aufbau- und Verbindungstechnik auf den Standardprozessen der Leiterplattentechnik basiert und deren Design Rules für den Entwurf der Kontaktstellen Anwendung gefunden haben, können auch die Erfahrungen aus diesem Bereich hinsichtlich mechanischer Stabilität und Zuverlässigkeit übertragen werden.

Positiver Nebeneffekt ist, dass dies zu einer geringeren Bauhöhe des Systems und somit zu einer noch kompakteren Bauweise führt. Bei einer Grundfläche von 12,5mm x 12,5mm waren typische Bauhöhen für verschiedene umgesetzte Systeme 12mm bis 25mm. Diese Systeme sind sowohl hinsichtlich ihrer kompakten Bauweise als auch bezüglich ihres mechanischen Aufbaus direkt in Anwendungen wie z.B. Maschinenkomponenten, Anlagen oder Versuchsaufbauten integrierbar.

Voraussetzung für eine Kompatibilität der EasyKit Baugruppen ist, dass die außen liegenden Kontakte nicht nur hinsichtlich ihrer Geometrie und Lage, sondern auch ihrer elektrischen Belegung definiert sind. Da diese Belegung einen herstellerübergreifenden Standard darstellen muss, wurde das VDMA Einheitsblatt 66305 hinsichtlich dieser Belegung für Baugruppen der Größe 12,5mm x 12,5mm erweitert. Die Arbeiten hierfür und die Schnittstellenbelegung sind in Kapitel 6.1 dargestellt.

Von großer Bedeutung ist, dass die in Stufe 1 erstellte Software ohne Änderungen weiterverwendet werden kann. Durch die Verwendung der identischen Funktionsträger finden keine Änderungen an der Schaltung statt, alle Bauelemente und ihre Verknüpfungen bleiben unverändert erhalten.

Somit ist ein Übergang in der Entwicklungsmethodik von Stufe 1 zu Stufe 2 möglich, ohne dass in der funktionswichtigen Hardware oder der Software Änderungen erforderlich sind. Bei vorliegenden Funktionsträgern kann der Übergang von Stufe 1 zu Stufe 2 innerhalb kürzester Zeit (bei Vorliegen der Funktionsträger innerhalb weniger Minuten, bei Bestellungen ab Lager innerhalb weniger Arbeitstage) erfolgen. Spezifisches Expertenwissen in den Bereichen Mechanik, Elektronik oder Informatik ist für diesen Übergang nicht erforderlich. Die Aufbau- und Verbindungstechnik ist so gestaltet, dass sie ohne umfangreicheres Erfahrungswissen anwendbar ist.

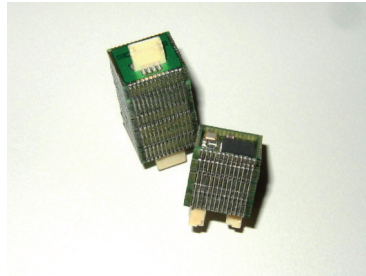
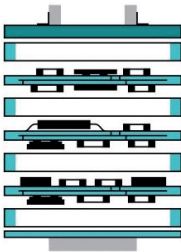


Abbildung 5.18: Systemaufbau in Stufe 2

Ein weiterer Vorteil beim Übergang zu Stufe 2 ist, dass eine derartig aufgebaute EasyKit Baugruppe (siehe Abbildung 5.19) auch wie ein Standardbauelement (vergleichbar einem MLF-Packaging) auf Leiterplatten bestückt werden kann. Ein weiterer Vorteil ist, dass insbesondere bei der Entwicklung von Schaltungen in kleinen Stückzahlen auf getestete Funktionsgruppen zurückgegriffen werden kann. So kann z.B. die CPU-Baugruppe als abgeschlossener Funktionsträger oder Baustein auf einen Schaltungsträger platziert werden, während die anderen Schaltungsteile anwendungsspezifisch layoutet und platziert werden.

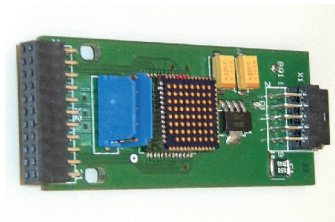


Abbildung 5.19: EasyKit Baugruppe auf Leiterplatte bestückt

Abschließend ist festzustellen, dass für Stufe 2 eine Kompatibilität der einzelnen Aufbauformen 1 bis 3 über das standardisierte Grid-Array im Raster 1,27mm gegeben ist, d.h. Bausteine mit außen liegenden Kontakten können z.B. auch mit Bausteinen in Starr-Flex-Technologie kombiniert werden. In der Praxis ist zu beachten, dass bei mehreren Lötritten ggf. abgestufte Lote zu verwenden sind. Hierfür wurden allerdings keine weiterführenden Untersuchungen durchgeführt.

6 Werkzeuge der Entwicklungsmethodik EasyKit

Die Entwicklungsmethodik EasyKit basiert auf unterschiedlichen Werkzeugen und Hilfsmitteln (siehe Abbildung 4.1), die Anwendern den Zugang erleichtern [Bauer2008]. Die Forderungen an diese Werkzeuge sind demzufolge ein einfacher Zugriff, eine gute Verständlichkeit und die Anwendbarkeit mit typischen und einfach verfügbaren Fertigungsmitteln. Die Werkzeuge wurden entwickelt, aufgebaut und in Anwendungen getestet. Im Folgenden sind diese kurz dargestellt.

6.1 Schnittstellendefinition

6.1.1 Anpassung vorhandener Schnittstellen

Um die Kompatibilitäten von Komponenten, Funktionsträgern und Bausteinen unterschiedlicher Hersteller und Entwickler sicherzustellen, wurde eine herstellerübergreifende Definition der Schnittstellen angestrebt. Da mit dem VDMA Einheitsblatt 66305, Schnittstellen der Mikrosystemtechnik, bereits eine umfassende und passende Basis vorliegt, erfolgte eine Ergänzung dieses Einheitsblatts um die für EasyKit erforderlichen Schnittstellen. Für die Stufen 1 und 2 der Entwicklungsmethodik EasyKit sind die bereits definierte geometrische und elektrische Schnittstelle von großer Bedeutung.

Die geometrische Schnittstelle beschreibt die Grundabmessung und Form der Funktionsträger bzw. Bausteine. Diese haben eine quadratische Grundform, die in den Abmessungen 5mm bis 25mm Kantenlänge mit einer Abstufung von 2,5mm variieren kann. Die Größe von 12,5mm wurde als Vorzugsgröße definiert. Die geometrische Schnittstelle hat Gültigkeit in den Stufen 1 und 2 des Entwicklungskonzepts. Eine Anpassung der bereits vorliegenden Schnittstellendefinition des Einheitsblatts war für die Entwicklungsmethodik EasyKit nicht erforderlich.

Die im Einheitsblatt definierte elektrische Schnittstelle hat Gültigkeit bei Stufe 1 des Entwicklungskonzepts und beschreibt sowohl die Lage der elektrischen Anschlüsse als auch deren Belegung bei den mikrotechnischen Bausteinen. Die elektrische Schnittstelle befindet sich auf der Unter- und Oberseite des Bausteins und ist ausgeführt als Grid Array im Raster von 1,27mm. Die Anschlüsse können als Kontaktflächen oder als Verbindungselement (Steckverbindung) ausgeführt sein. Darüber hinaus sind die Anschlüsse auch hinsichtlich ihrer Signalbe-

legung definiert. Bei der Vorzugsgröße von 12,5mm sind 63 Anschlüsse im Raster 8x8-1 angeordnet. Eine Anpassung dieser Definition ist nicht erforderlich.

Über die elektrische Schnittstelle ist auch die Belegung dieser Anschlüsse definiert (Abbildung 6.1). Diese Anschlussbelegung unterteilt die Anschlüsse in erster Linie in digitale und analoge Pins und weist diesen eine Vorzugsbelegung zu (siehe auch Abbildung 3.7). Das Problem an dieser Belegung war, dass die meisten Anschlussdefinitionen nicht sehr konkret waren und durch freie Belegungsmöglichkeiten der Anwender schnell zu Inkompatibilitäten geführt haben.

Um einerseits eine Kompatibilität von Bausteinen unterschiedlicher Hersteller sicherzustellen und andererseits kein zu starres System zu definieren, wurde für die Entwicklungsmethodik EasyKit die Schnittstellenbelegung konkreter gefasst und in vier Kategorien unterteilt:

Kategorie 1: Definierte Anschlussbelegungen

In Kategorie 1 erfolgen verbindliche Anschlusszuweisungen, die bei allen Bausteinen einzuhalten sind. Dies betrifft die Anschlüsse für die Spannungsversorgung (UA, UD, GA, GD) und die Kommunikation (RXD, TXD, CANH, CANL). Die Anschlüsse sind bei aktiven Elementen mit diesem Signal zu belegen oder bei passiven Elementen zwingend durchzuführen.

Kategorie 2: Vorzugsbelegungen

Die Vorzugsbelegungen gewährleisten für übliche Aufbauformen die Kompatibilität. Die Vorzugsbelegung beschreibt eine prinzipielle Unterteilung in Digital- und Analoganschlüsse sowie für einzelne Anschlüsse deren Belegung. Auf digitaler Seite sind dies die Anschlüsse für die interne Kommunikation (I²C, SPI) und für digitale Ein- und Ausgänge. Auf analoger Seite werden Anschlüsse für Spannungs- und Stromsignale (Ein- und Ausgänge) definiert wie auch Anschlüsse, die für intern durchzuleitende Signale Verwendung finden.

Kategorie 3: Freie Anschlüsse / Empfehlung (Bus mit 52 Anschlüssen)

Diese Anschlüsse sind für anwendungsspezifische Funktionen frei verfügbar. Unterbrechungen der Leitungen oder Signalwechsel sind möglich. Eine Empfehlung für diese Anschlüsse ist angegeben, sollten diese in der jeweiligen Bausteinumsetzung nicht mit anderen Signalen belegt sein wird empfohlen, die hier angegebenen Signale zu verwenden.

Kategorie 4: Freie Anschlüsse / Empfehlung (Bus mit 63 Anschlüssen)

Diese Anschlüsse sind für anwendungsspezifische Funktionen frei verfügbar. Die Anschlüsse sind nur bei Verwendung einer Bauform mit 63 Anschlüssen (z.B. Starr-Flex-Bausteine) zu verwenden.

	5	6	7	8	9	10	11	12
E		MS1	MS2	MS3	MS4	RB6	RB7	GD2
F	PRG	RES	CS	RC0	NPN	SD0	SDI	UD2
G	IDL_1	INT	GD1	NC	NC	UD1	SCK	IDL_2
H	NC	CANL	TXD	NC	NC	RXD	CANH	NC
J	U _{IN} 24	U _{IN} 24	NC	NC	NC	NC	NC	PNP _{THR}
K	RA0	AN _{IN} 2	UA1	AN _{IN} 1	AN _{OUT} 2	GA1	GD _{EXT} 1	GD _{EXT} 2
L	RA1	LA _{OUT} 2	URH	AN _{ADC} 1	AN _{ADC} 3	URL	AN _{OUT} 1	RA4
M	RA2	AN _{IN} 3	GA2	AN _{ADC} 2	AN _{ADC} 4	UA2	LA _{OUT} 1	RA5

Kategorie 1
 Kategorie 2
 Kategorie 3
 Kategorie 4

Abbildung 6.1: Erweiterte Anschlussbelegung der elektrischen Schnittstelle am Beispiel eines Bausteins der Größe 12,5mm

6.1.2 Erweiterung um die elektromechanische Schnittstelle

Für die Stufe 2 der Entwicklungsmethodik EasyKit, die Verbindung von Funktionsträgern, hat im bisherigen VDMA Einheitsblatt 66305 eine Definition gefehlt. Daher wurden die bereits vorhandenen Definitionen um die elektromechanische Schnittstelle erweitert. Die elektromechanische Schnittstelle beschreibt die direkte Verbindung zwischen Funktionsträgern. Dabei stellt sie eine Verbindung zwischen den mechanischen Anforderungen (Aufbau Rahmen, Höhe, mechanische Verbindung der einzelnen Ebene) als auch den elektrischen Anforderungen (Signalführung, Belegung des Anschlusses) dar (Abbildung 6.2). Die Definition gilt derzeit für Funktionsträger der Größe 12,5mm x 12,5mm.

An jeder Außenseite der quadratischen Grundfläche befinden sich 13 Anschlüsse mit einem Rastermaß von 0,8mm, d.h. in Summe 52. Diese Anschlüsse sind durch das Element geführt. Die elektrischen Anschlüsse haben auf der Ober- und Unterseite des Funktionsträgers bzw. der Komponente jeweils eine Kontaktfläche, über die anschließend die Verbindung mittels Lötens hergestellt werden kann. Darüber hinaus sind als Montagehilfe in den Ecken Bohrungen mit einem Durchmesser von 0,6mm bzw. 0,8mm definiert, die insbesondere bei der manuellen Montage für Funktionsmuster oder im Prototypenbau die Fertigung vereinfachen.

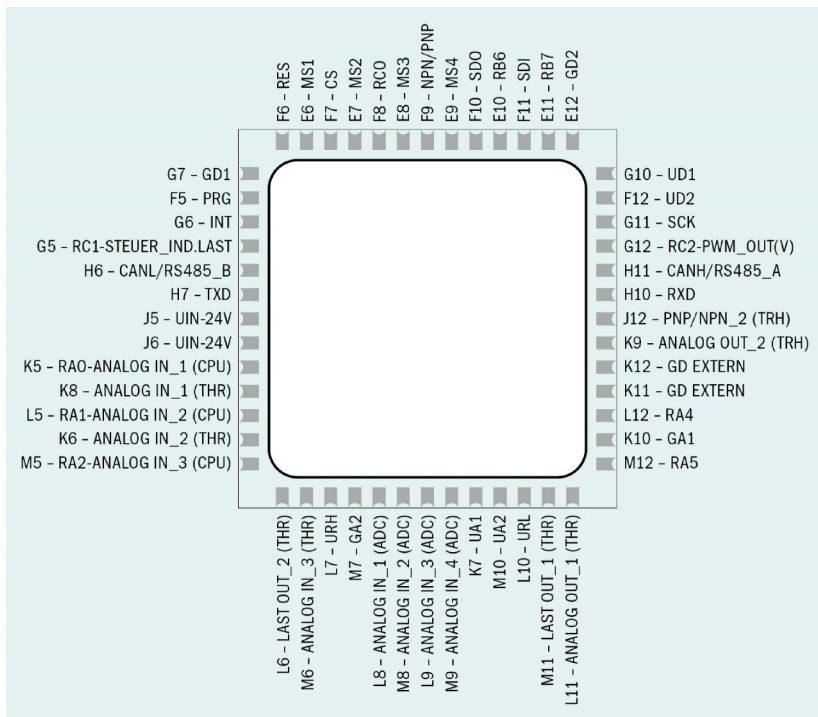


Abbildung 6.2: Belegung der elektromechanischen Schnittstelle

Wie schon bei der elektrischen Schnittstelle liegt eine Zuweisung der elektrischen Anschlussbelegung vor (siehe Abbildung 6.2). Wie in der elektrischen Schnittstelle beschrieben erfolgt auch hier die Unterteilung in die drei Kategorien (Definierte Anschlussbelegungen, Vorzugsbelegung, freie Belegung).

Für den Übergang der elektrischen Schnittstelle (Grid Array) auf die elektromechanische Schnittstelle (außen liegender Bus) liegt ein Vorschlag zur Umverdrahtung vor, der von Dritten übernommen werden kann.

6.2 Funktionsträger, Komponenten und Bausteine

6.2.1 Funktionsträger

Mit dem Begriff Funktionsträger werden elektronische und mechatronische Baugruppen definiert, die eine in sich geschlossene Einzelfunktion darstellen und in Kombination mit anderen Funktionsträgern eine übergeordnete Gesamtfunktion erfüllen können. Die Funktionsträger verfügen über eine quadratische Grundform mit den in Kapitel 6.1 definierten Schnittstellen, so dass diese einfach untereinander kombiniert werden können. Für die Fertigung können unterschiedliche Technologien zum Einsatz kommen. Die Fertigung erfolgt wie bereits beschrieben auf Basis der etablierten Leiterplattentechnologie. Prinzipiell können diese Funktionsträger aber auch in anderen Technologien wie z.B. der LTCC- oder der MID-Technologie mit bereits integrierten Rahmenelementen hergestellt werden [SCHÜNEMANN 2003]

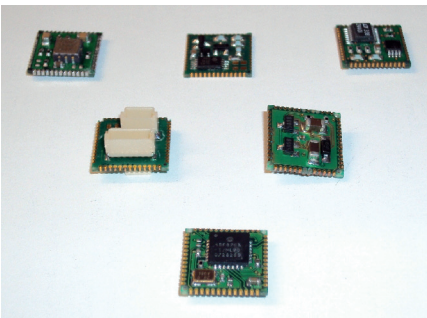


Abbildung 6.3: Verschiedene Ausführungen von Funktionsträgern

Die Funktionsträger lassen sich wahlweise in Verbindung mit Rahmen, Deckelelementen und/oder Steckerelementen zu einzelnen Bausteinen komplettieren (Bausteine für Stufe 1 des Entwicklungskonzepts) oder können durch direkte Kombination inklusive der erforderlichen Rahmen zu anwendungsspezifischen Systemen (Stufe 2) aufgebaut werden. Außerdem können die einzelnen Funktionsgruppen auch als eigenständige Baugruppe auf Leiterplatten zur Anwendung kommen.

Beispiele für elektronische oder mechatronische Funktionsgruppen sind z.B.:

- Signalverarbeitende Funktionen (beschaltete Mikroprozessorbaugruppen)
- Signalwandelnde Funktionen (A/D-Wandler oder D/A-Wandler mit entsprechender Beschaltung)
- Funktionen zur Spannungsanpassung
- Treiberstufen
- Sensor- und Aktorbaugruppen

Die interne Ausführung der Baugruppe obliegt dabei grundsätzlich dem Entwickler oder Hersteller. Für eine Einbindung in die Entwicklungsmethodik EasyKit ist lediglich wichtig, dass die in Kapitel 5.1 beschriebenen Schnittstellen (geometrisch und elektromechanisch) eingehalten werden.

6.2.2 Komponenten

Mit dem Begriff Komponenten werden Bauelemente definiert, die zum Aufbau von Bausteinen oder von kompletten Systemen erforderlich sind. Im Einzelnen fallen hierunter z.B.:

- Rahmenelemente
- Deckel- und Bodenelemente
- Interfaceelemente (Stecker und Verbindungselemente)

Die Komponenten verfügen über definierte Schnittstellen, die je nach Ausführung einen Aufbau in Stufe 1 und/oder 2 unterstützen.

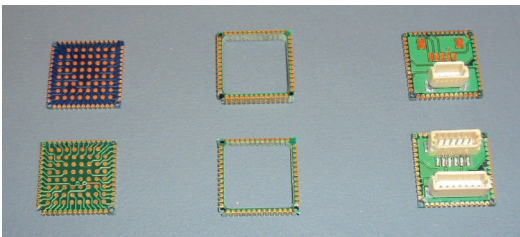


Abbildung 6.4: Komponenten (v.l.n.r.: Deckel/Boden, Rahmen, Interfaces)

6.2.3 Bausteine

Bausteine sind in sich geschlossene Funktionsgruppen, die durch Kombination mehrerer Bausteine eine Gesamtfunktion erfüllen und so den direkten Aufbau von Systemen ermöglichen. Der Baustein selbst besteht aus mehreren Ebenen (Funktionsträger und Komponenten), wobei der innere Aufbau jedem Hersteller frei gegeben ist. Die elektrischen Anschlüsse sind auf einen internen Bus (= elektromechanische Schnittstelle) geführt, der durch den kompletten Baustein führt. Den Abschluss des Bausteins bildet jeweils eine Boden- und Deckplatte. Diese wiederum verfügen über die definierte elektrische Schnittstelle an der Ober- und Unterseite. Die Anschlüsse können in Form von Steckern oder in Form von Kontaktierungsflächen ausgeführt sein.

Die steckbare Variante eröffnet eine einfache, direkte Kontaktierung der Bausteine, die sich beliebig häufig verbinden und auch wieder öffnen lässt. Somit können vorhandene Bausteine auch in unterschiedlichen Konfigurationen zusammengesetzt und anschließend wieder auseinander genommen werden.

Die Bausteine sind insbesondere für die Stufe 1 des Entwicklungskonzepts ausgelegt.

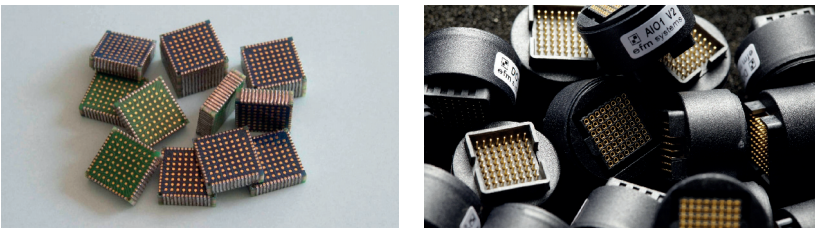


Abbildung 6.5: Bausteine (Links mit Grid-Array, Rechts mit Federmatrix)

Der Anwender hat somit die Möglichkeit, ohne Kenntnisse der konkreten Schaltung und deren Details auf Basis der Funktionsbeschreibung und der Schnittstellenbelegung die Eignung für die geforderte Gesamtfunktion und die Kompatibilität mit anderen Funktionsgruppen zu prüfen.

6.3 Technologie und Entwicklungsprozess am Beispiel eines Beschleunigungsmesssystems

Da in einem Baukastensystem nicht alle erforderlichen Funktionen für jede beliebige Anwendung zur Verfügung stehen können, ist es erforderlich, dem Anwender im Bereich der Entwicklung und Fertigung Hilfsmittel zur Verfügung zu stellen, die sowohl im Labormaßstab als auch für die Kleinserienfertigung umfangreiche Unterstützung zur eigenständigen Ergänzung dieser Funktionen bieten und vor allem den Zugang zu diesem System vereinfachen.

In aller Regel werden beim Aufbau eines mechatronischen Systems auf Basis eines Baukastens nicht alle erforderlichen Funktionen verfügbar sein. So sind z.B. in den Bereichen Spannungsversorgung, Kommunikation und Signalverarbeitung die meisten Funktionen in unterschiedlichsten Systemen einsetzbar, eine hohe Verfügbarkeit und Wiederverwendbarkeit an bereits vorhandenen Funktionsgruppen ist somit gegeben. Je spezifischer jedoch die Anforderungen werden (z.B. sensorspezifische Verstärker, Filter) nimmt diese Verfügbarkeit deutlich ab. Daher wurden mit der Entwicklungsmethodik EasyKit Möglichkeiten geschaffen, diese Funktionsblöcke einfach und effizient zu entwickeln (Unterstützung bei der Schaltungsentwicklung), als Funktionsträger umzusetzen (Layoutunterstützung) und diese letztendlich aufzubauen (Fertigungsunterstützung) oder zu testen (Testwerkzeuge). So ist es möglich, eigene Entwicklungen aus vorhandenen Funktionsblöcken und aus selbst erstellten Funktionsblöcken zu kombinieren, ohne in jedem dieser Teilgebiete über tiefer gehendes Expertenwissen zu verfügen.

Um diese Schritte zu demonstrieren, wird in den nächsten Kapiteln jeweils begleitend die Entwicklung eines Beschleunigungsmesssystems unter Nutzung der Entwicklungsmethodik EasyKit beschrieben. Der Fokus liegt dabei nicht auf dem Entwurfsablauf, sondern auf den begleitenden Werkzeugen und deren Beschreibung.

An das Messsystem werden folgende grundlegende Anforderungen gestellt:

- Erfassung der Erdbeschleunigung in 2-Achsen
- Messbereich: $\pm 2g$
- Messauflösung: 10mg
- Signalaufbereitung vor Ort
- Signalausgabe über RS485/Modbus als einheitenrichtiger Wert

6.3.1 Funktionsstruktur und Teilfunktionen

Grundvoraussetzung für die Schaltungsentwicklung ist die Kenntnis über die erforderlichen Funktionen, die Ein- und Ausgänge des Systems sowie die Teilfunktionen. Somit ist in einem ersten Schritt die Gesamtfunktion eines Systems in die Einzelfunktionen zu zerlegen.

Für diese Einzelfunktionen sind die bereits im Baukasten vorhandenen Funktionsblöcke zu ermitteln. Diese Funktionsblöcke sind anhand einfacher Blockschaltbilder sowie zugehöriger Datenblätter beschrieben. Dieses Vorgehen entspricht gleichzeitig den ersten Schritten des Entwurfsablaufs nach der Entwicklungsmethodik EasyKit (Systementwurf – siehe Abbildung 4.4).

Anhand des Beispiels Beschleunigungsmesssystem wird im Folgenden das prinzipielle Vorgehen beschrieben:

Schritt 1 – Zerlegung des Systems in Teilfunktionen (siehe Abbildung 6.6)

- Teilfunktion 1: Signalerfassung
Die Erfassung der Beschleunigung erfolgt über ein Sensorelement.
- Teilfunktion 2: Signalaufbereitung und Signalwandlung
Das erfasste Sensorsignal wird bei Bedarf aufbereitet (z.B. verstärkt und gefiltert), gewandelt und aus dem analogen Sensorsignal wird ein für eine nachfolgende Einheit verwertbares digitales Signal.
- Teilfunktion 3: Signalverarbeitung
Die Berechnung der Beschleunigung erfolgt aus dem vorliegenden Sensorsignal unter Einbeziehung der erforderlichen Korrektur- und Kalibrierfunktionen, die als Funktionen hinterlegt sind.
- Teilfunktion 4: Kommunikation
Dies umfasst sowohl die datentechnische Aufbereitung der Daten in Form des erforderlichen Busprotokolls (hier Modbus) als auch die Bereitstellung der entsprechenden Treiber für die entsprechenden RS485 Pegel.
- Teilfunktion 5: Spannungsaufbereitung
Während dem System von außen eine Spannung im Bereich von 10...35V DC zur Verfügung gestellt werden kann, arbeitet das System intern mit einer geringeren Spannung (typisch 3,3V oder 5V). Diese Adaption sowie eine rudimentäre Schutzbeschaltung und ggf. auch entsprechende EMV-Maßnahmen sind durch diese Teilfunktion sicherzustellen.

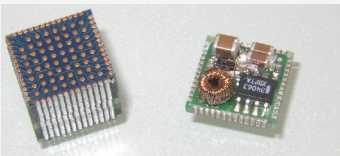


Abbildung 6.6: Teilfunktionen des Beschleunigungsmesssystems

Schritt 2 – Prüfung der verfügbaren Teilfunktionen

Zur Prüfung der verfügbaren Teilfunktionen (im Weiteren auch als Funktionsblock bezeichnet) steht innerhalb der Entwicklungsmethodik EasyKit ein Funktionskatalog zur Verfügung. Dieser beschreibt in Form eines Datenblatts (Abbildung 6.7) die grundlegenden Eigenschaften des Funktionsblocks und stellt in einem Blockschaltbild (Abbildung 6.8) dessen Schnittstellen dar.

Baustein zur Spannungsversorgung – Reglerbaustein SRM2.0



Baustein

Eingangsspannung	UE = +10V bis +35V DC
Ausgangsspannung	UA = +5V UD = +5V
Strom	IA = Typ. 100mA (max. 200 mA) ID = Typ. 100mA (max. 200mA) Zusammen max. 200mA
Genauigkeit	Typ. 1%UA bzw. DU

Spezifikationen

Restwelligkeit	<1mV @ 100mA (<2mV @ 200mA)
Wirkungsgrad	≥ 70% @ UE 24V, IA+ID = 200mA)

Schaltfrequenz	30-100 kHz
Temperatur-Koeffizient	Typ. 100ppm/°C (max. 250ppm/°C)
Kurzschlussicherung durch Strombegrenzung	IA+ID = max. 300mA
Baustein	
Temperaturbereich	-40°C...+70°C
Größe	12,5 x 12,5 x 7,2mm ³
Kompatibilität	0.63 (Pinbelegung siehe Datenblatt)
Kontaktierung	Top: Land Grid, (3µm NiAu) Bottom: Land Grid (3µm NiAu)
Material	FR4
Farbmarkierung	Schwarz / Weiß

Abbildung 6.7: Datenblatt des Funktionsblocks Spannungsaufbereitung der EasyKit Entwicklungsmethodik

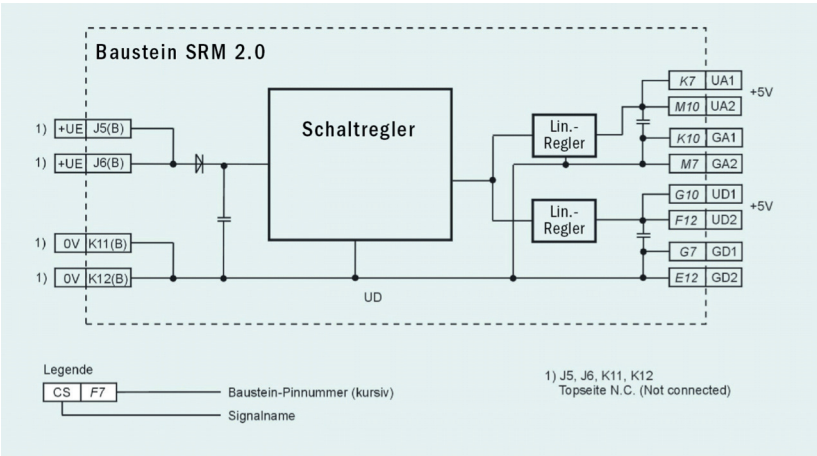


Abbildung 6.8: Blockschaltbild der Funktion Spannungsaufbereitung

Die interne Schaltung der Funktion bleibt hingegen Wissen des Herstellers und kann, aber muss nicht zwingend offen gelegt werden. Durch Kombination unterschiedlicher Funktionsblöcke kann so die Realisierung der geforderten Gesamtfunktion geprüft werden. Im vorgenannten Beispiel des Beschleunigungsmesssystems können aus den vorhandenen Funktionsblöcken die Teilfunktionen 3 (Signalverarbeitung), 4 (Kommunikation) und 5 (Spannungsaufbereitung) erfüllt werden. Für die Teilfunktion 1 (Signalerfassung) steht kein Funktionsblock zur Verfügung. Die Teilfunktion 2 (Signalwandlung) ist verfügbar, jedoch ist zu prüfen, ob die Anforderungen für ein noch zu bestimmendes Sensorelement erfüllt werden können oder erforderlich sind.

Schritt 3 – Erstellung eines eigenen Funktionsblocks

Da für die Signalerfassung kein Funktionsblock vorliegt, ist die Erstellung eines eigenen erforderlich. Nach einer Recherche wurde für die Anforderungen an das Beschleunigungsmesssystem der Sensor ADXL202 [ANALOG 2008_ADXL] gewählt. Dieser 2-achsige Beschleunigungssensor verfügt über eine integrierte Signalvorverarbeitung. Das Signal wird als PWM zur Verfügung gestellt. Somit muss für die nachfolgende Signalverarbeitung nur der entsprechende Anschluss beachtet werden.

6.3.2 Schaltungsentwicklung

Die Erstellung des Schaltplans berücksichtigt somit eingangsseitig nur den Beschleunigungssensor, der ausgangsseitig mit den entsprechenden Anschlüssen am Signalverarbeitungsmodul zu verbinden ist. Darüber hinaus wird ein interner Speicher vorgesehen, über den sofort auch die Kalibrierdaten des Beschleunigungssensors dauerhaft gespeichert werden. Dieser wird mit einer SPI-Schnittstelle ausgeführt und ermöglicht so auch wieder den direkten Anschluss an das Signalverarbeitungsmodul. Des Weiteren ist die Spannungsversorgung zu berücksichtigen, die von der Spannungsaufbereitung zur Verfügung gestellt wird. Im Schaltplan muss der Entwicklungsmethodik Rechnung getragen werden, indem auf die entsprechende Pinkompatibilität und Nomenklatur [VDMA 2003] geachtet werden muss. Hier ist z.B. die Spannungsversorgung für analoge Komponenten auf den Anschlüssen K7 (UA) und K10 (GDA) zu führen.

Weiterhin ist zu berücksichtigen, dass seitens der Entwicklungsmethodik eine Authentifizierung der Funktionsblöcke vorgeschlagen wird (Modulselect = MS). Hierfür stehen zwei Möglichkeiten zur Verfügung, eine analoge Auswahllogik

über die im Bus definierten MS-Pins (MS1 bis MS4) oder als zweite Variante durch Integration eines kleinen Controllers, in dem eine Authentifizierungsroutine hinterlegt ist. Im vorliegenden Fall des Beschleunigungssensors wurde die analoge Auswahllogik eingesetzt, wobei die Pins MS1 bis MS3 wahlweise über eine Brücke angeschlossen werden können. Dies würde beispielsweise auch die Möglichkeit eröffnen, mehrere Sensorbausteine gleichzeitig zu betreiben und über die MS Pins zu initialisieren.

Im Ergebnis liegt der Schaltplan für das Beschleunigungsmessmodul vor (siehe Abbildung 6.9).

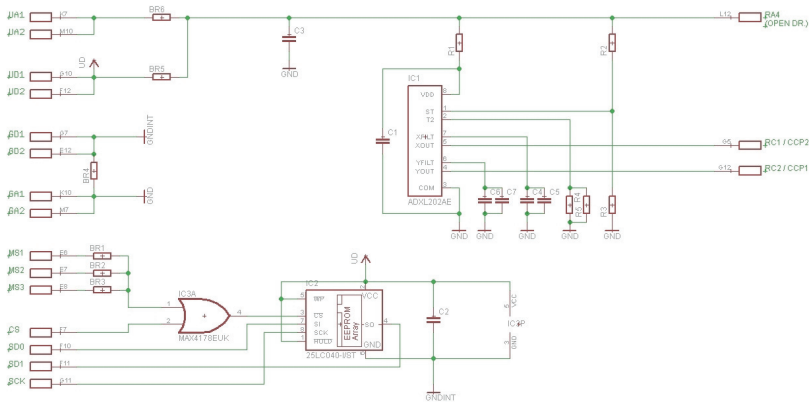


Abbildung 6.9: Schaltplan des Beschleunigungsmessmoduls

6.3.3 Layout

Im nächsten Schritt ist es erforderlich, aus dem nun verfügbaren Schaltplan ein Layout zu erstellen, welches die Kompatibilität mit den anderen Funktionsträgern ermöglicht. Dieses Layout muss neben der eigentlichen Umsetzung des Schaltplans (Grundfunktion + Verdrahtung auf die entsprechenden Anschlüsse des Bussystems) auch die richtige Anschlussgeometrien für den Funktionsträger und die Anschlüsse enthalten.

Für die Entwicklungsmethodik EasyKit wurde eine Layoutvorlage entwickelt, die dem Anwender diese Schnittstelle zur Verfügung stellt, so dass eine Vereinfachung der Arbeiten erfolgen kann, indem nur der Schaltplan in einen vorgegebene Rahmen implementiert werden muss. Die Layoutvorlage existiert einerseits

als technische Zeichnung inkl. Beschreibung der Anschlussbelegungen, andererseits als Vorlage in einem Layoutprogramm. Hier wurde das Programm Eagle [EAGLE 2008] gewählt, da dieses im Laborbereich und unter Entwicklern eine sehr weite Verbreitung findet. Die Layoutvorlage enthält die definierte äußere Struktur des Funktionsträgers (Vias mit entsprechenden Restringen) sowie die Zuweisung der Schnittstellen. Der Anwender kann im freien Bestückungsraum (zweiseitig 10,5mm x 10,5mm, d.h. 220,5mm² Bestückungsfläche) seine eigene Schaltung/Bauelemente platzieren und diese entsprechend seinen Anforderungen verdrahten. Des Weiteren enthält die Layoutvorlage auch die Fräskonturen, um die außen liegende Kontaktierung zu erzeugen (siehe Abbildung 6.10). Hierfür werden die Vias mittig getrennt und kleine Stege stehen gelassen, die anschließend ein einfaches Vereinzeln ermöglichen.

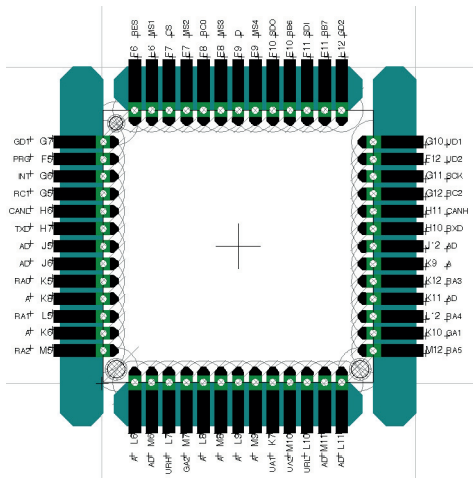


Abbildung 6.10: Layoutvorlage für einen Funktionsträger

Für das Beispiel des Beschleunigungsmesssystems sind die Bauelemente der Schaltung des Funktionsmoduls Beschleunigungssensor (Abbildung 6.9) auf der Bestückungsfläche von 220mm² unterzubringen. Es wurde ein Design gewählt, bei dem auf der Oberseite der Sensor mit seiner Beschaltung (analoger Schaltungsteil) und auf der Unterseite der EEPROM mit den Kalibrierdaten und der Logik für die Auswahl des Moduls (digitaler Teil) integriert wurde. Im Ergebnis entstand ein 4-lagiger Aufbau.

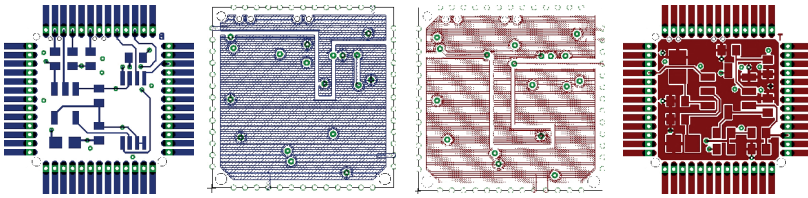


Abbildung 6.11: Layout Funktionsmodul Beschleunigungssensor

6.3.4 Leiterplattenfertigung und -bestückung

Nach Abschluss des Layouts sind die Leiterplatten zu fertigen und zu bestücken. Durch Verwendung gängiger Leiterbahnbreiten, -abstände und Bohrdurchmesser können die Leiterplatten kostengünstig gefertigt werden. Ausnahme hierbei ist der Restring der Vias des außen liegenden Bussystems. Dieser ist mit $100\mu\text{m}$ dimensioniert. Die meisten Leiterplattenhersteller gehen hier von einem Standardmaß von $120\ldots150\mu\text{m}$ aus. Für $100\mu\text{m}$ ist typischerweise mit einem Aufschlag von 10% zu rechnen. Allerdings lässt sich dieser Restring nicht größer dimensionieren, da ansonsten die Abstände zum benachbarten Restring ein kritisches Maß unterschreiten und so die Gefahr von Kurzschlüssen entsteht.

Die Anordnung des Nutzens ist natürlich jedem Anwender freigestellt. Um jedoch auch hier einerseits dem Anwender eine Unterstützung zu geben, andererseits die Montage kompletter Nutzen zu ermöglichen wurden wiederum in der Entwicklungsmethodik EasyKit Vorgaben und Vorlagen erstellt.

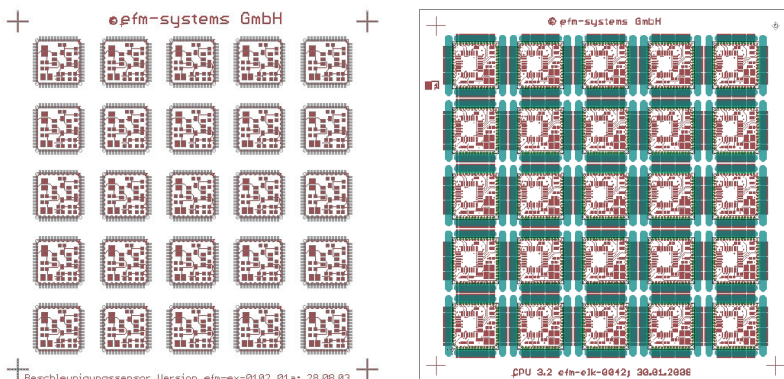


Abbildung 6.12: Layout eines Nutzens (links ohne Kantenfräsung, rechts mit Kantenfräsung)

Als Empfehlung für die Anordnung des Nutzens wurde eine Anordnung mit 5x5 Funktionsträgern gewählt. Die Funktionsträger sind in einem Rastermaß von 17,5mm angeordnet. Ein Nutzen hat somit eine typische Größe von 100mm x 100mm. Es liegt wiederum sowohl eine technische Zeichnung als auch eine Layoutvorlage vor (siehe Abbildung 6.12). Diese Nutzenanordnung ermöglicht somit eine vereinheitlichte Herstellung der Komponenten und Funktionsträger. Im Rahmen der Entwicklung wurden verschiedene Leiterplattenhersteller getestet. Im Ergebnis entstand für die Entwicklungsmethodik EasyKit eine Liste mit empfohlenen Leiterplattenherstellern.

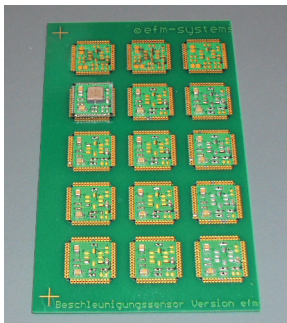


Abbildung 6.13: Manuell teilbestückte Leiterplatte im Nutzen

Die Nutzen sind so ausgelegt, dass sie sowohl für eine maschinelle als auch für eine manuelle Weiterverarbeitung geeignet sind. Im Prinzip können alle gängigen Technologien (SMD, THT, FlipChip, COB) zum Einsatz kommen, in der Praxis findet jedoch fast ausschließlich die SMD-Technologie Verwendung. Diese hat den Vorteil einer breiten Verfügbarkeit der Bauelemente in allen Stückzahlbereichen sowie einer einfachen Verarbeitbarkeit, wiederum sowohl manuell als auch maschinell. Für die Prototypenherstellung können die Funktionsträger einzeln manuell per LötKolben bestückt werden. Eine gewisse praktische Erfahrung im Löten ist hier jedoch vorauszusetzen.

Für die maschinelle Fertigung werden die kompletten Nutzen bestückt (siehe Abbildung 5.5). Hier kann durch die gewählten Technologien auf bewährte Fertigungsprozesse und -abläufe der Leiterplattenbestückung zurückgegriffen werden. Die Bestückung ist mit jedem gängigen Halb- oder Vollautomaten möglich. Bei Bedarf können bereits während des Standardbestückungsprozesses die Rahmen auf Ober- und Unterseite mit bestückt und verlötet werden. Dies verein-

facht die Weiterverarbeitung, erhöht die Zuverlässigkeit und senkt die Kosten für die Herstellung eines kompletten Bausteins. Derartig vorgefertigte Funktionsträger können sowohl in Stufe 1 als auch in Stufe 2 des Entwicklungskonzepts genutzt werden.

Beim Beispiel des Beschleunigungsmesssystems kommen ausschließlich SMD-Komponenten zum Einsatz. Aufgrund des zur Verfügung stehenden Bauraums wurde bei Widerständen und Kondensatoren als kleinste Chip-Bauform die Größe 0603 gewählt, um eine einfachere manuelle Bestückung zu ermöglichen.

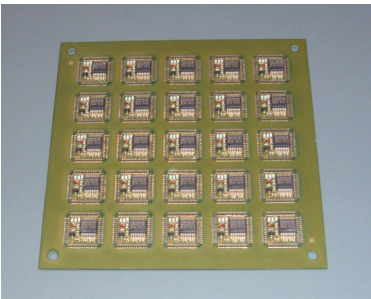


Abbildung 6.14: Maschinell bestückter Nutzen

6.3.5 Baustein- und Systemaufbau

Aus dem vorliegenden Funktionsträger kann nunmehr ein Baustein der Stufe 1 oder ein komplettes System (Stufe 2) gefertigt werden. Für den Aufbau eines Bausteins ist der Funktionsträger mit dem Komponenten Rahmen sowie Deckel und Boden zu komplettieren. Für das Beschleunigungsmesssystem benötigt der Baustein jeweils einen Rahmen auf Ober- und Unterseite. Deckel und Boden können je nach Bedarf als Grid Array oder mit Matrixstecker ausgeführt sein.

Wie in Kapitel 5.1.2 beschrieben und in Abbildung 5.4 dargestellt weisen alle Komponenten Montagebohrungen auf, die den Zusammenbau des Bausteins unterstützen sollen. Um den Anwendern im Prototypenbau diesen Zusammenbau zu erleichtern, wurde in der Entwicklungsmethodik EasyKit für die manuelle Montage ein Fertigungshilfsadapter konstruiert. Dieser verfügt über eine Basisplatte mit 3 Stiften (2 x 0,75mm, 1 x 0,55mm), auf die die einzelnen Komponenten aufgefädelt werden. Anschließend wird über eine Gegenplatte der Aufbau fixiert und über eine Spannschraube zusammengepresst. Dieser Adapter kann

in einen Schraubstock eingesetzt werden. Der Innenteil ist um 360° drehbar, so dass ohne weiteres Versetzen das Zusammenlöten des Bausteins möglich ist.

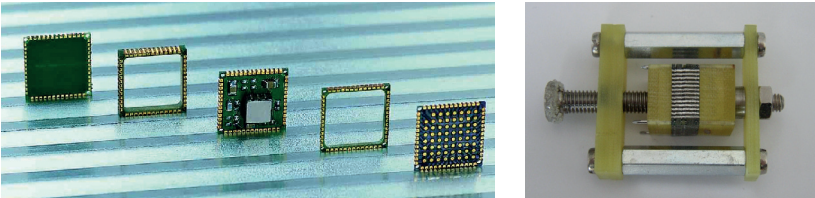


Abbildung 6.15: Erforderliche Komponenten zur Komplettierung des Bausteins (links) und Fertigungshilfsadapter mit Baustein (rechts)

Vor dem Löten empfiehlt es sich die aufgetrennten Vias auf Rückstände vom Fräsprozess zu kontrollieren. Anschließend sind die Kontaktstellen mit einem Flussmittel zu benetzen. Die Kontaktstellen können durch einfaches Entlangfahren mit dem LötKolben und der Zuführung einer entsprechenden Zinnmenge verbunden werden. Es ist darauf zu achten, nicht zuviel Zinn zu verwenden. Entscheidend für einen erfolgreichen Verbindungsprozess ist, dass das Lot nicht an den Kontaktbahnen der Halbvias entlangläuft, sondern in die Fläche mit den Restringen fließt, um dort auch eine mechanisch stabile Verbindung zu gewährleisten. Im Ergebnis liegen durchgehend verbundene Bahnen vor. Das Ergebnis kann zuverlässig unter dem Mikroskop kontrolliert werden (siehe Abbildung 6.16).

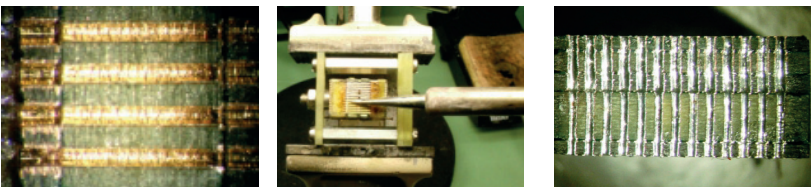


Abbildung 6.16: Gesäuberte Kontaktstellen vor dem Löten (links), Lötprozess (Mitte) und gelötte Seite eines Bausteins (rechts)

Auf gleiche Art und Weise kann auch ein komplettes System (Stufe 2 der Entwicklungsmethodik) aufgebaut werden. Hierzu sind die einzelnen Funktionsträger direkt mit Rahmenelementen zu verbinden. An der Ober- und Unterseite können abschließend Interfacebaugruppen angeschlossen werden, die eine Verbindung zu den Sensoren/Aktoren bzw. zur Versorgung und Kommunikation

sicherstellen. Auch für diesen Aufbau kann wiederum der Fertigungsadapter wie bereits beschrieben verwendet werden.

Für das Beispiel des Beschleunigungsmesssystems gilt, dass neben dem entwickelten Messmodul die Funktionsträger zur Spannungsversorgung, zur Signalverarbeitung und zur Kommunikation verbunden werden müssen. Darüber hinaus ist ein Interfacemodul zu integrieren, das den Anschluss zur Spannungsversorgung, zur Kommunikation und bei Bedarf auch zur Programmierung sicherstellt. Zwischen jeder Funktionsgruppe ist jeweils ein Rahmen, bei Bedarf aufgrund der Bauhöhe einzelner Bauelemente auch zwei Rahmen, angeordnet.

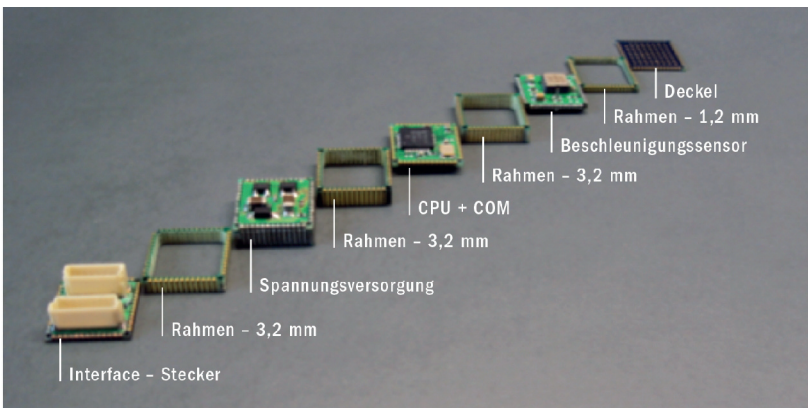


Abbildung 6.17: Funktionsgruppen des Beschleunigungsmesssystems in Stufe 2

Im Ergebnis entsteht ein Verbund aus einzelnen Elementen mit einer Gesamthöhe von ca. 15mm (ohne Bauhöhe der Stecker). Dieser wird in den Fertigungsadapter eingespannt. Das Löten des Systems ist identisch dem bereits Beschriebenen, es ist hier darauf zu achten, dass das Lot über die gesamte Länge der Bahnen gleichmäßig fließt und insbesondere in die Zwischenräume eindringt. Im Ergebnis liegt ein System in sehr kompakter Bauform vor, das in einem industrieüblichen Gehäuse (siehe Abbildung 6.18) integriert werden kann, Abmessungen von $\varnothing 20\text{mm} \times 45\text{mm}$ aufweist und für den praktischen Einsatz geeignet ist.

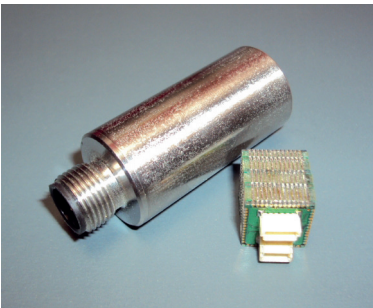
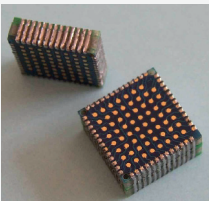


Abbildung 6.18: Beschleunigungsmesssystem im Edelstahlgehäuse mit Durchmesser von 20mm

6.3.6 Bausteindokumentation

Für den neu erstellen Funktionsträger bzw. Baustein ist analog zu den bereits vorhandenen ein Datenblatt und ein Blockschaltbild zu erstellen, um diese auch für weitere Anwendungen verfügbar zu machen. Hierfür können vorhandene Vorlagen verwendet werden.

Baustein zur Signalwandlung – Beschleunigungssensor AXY 2.1 und 2.2



Sensorelement

Achsen	2-achsiger Beschleunigungssensor
Messgröße	Beschleunigung [g], 2-achsiger (x, y)
Messbereich	$\pm 2g$ max. Beschleunigung (Schock): 1000g max. Beschleunigung (bei 5V): 500g (0,5ms)

Messbandbreite	AXY2.1: $f_c = 0\text{-}12\text{Hz}$ @ Auflösung 0,8mg [rms] AXY2.2: $f_c = 0\text{-}110\text{Hz}$ @ Auflösung 2,5mg [rms] Sonderausführungen möglich bis : $f_c = 0\text{-}500\text{Hz}$ @ Auflösung 5,7mg [rms]
Auflösung	2mg (Noise Density @ 25 °C $1000\mu\text{g} \cdot \sqrt{\text{Hz}}$ [rms] max.
Baustein-Schnittstellen	
Kommunikation intern	SPI-Bus (EEPROM-Daten)
Messgrößen	
XOUT	Duty-cyclus modulation
YOUT	Duty-cyclus modulation
Kalibrierung	Selbsttest über Anschluss zu Open Drain/Collector Normaloperation LOW (Selbsttest offen @ 0,8g)
Baustein	
Spannungsversorgung	+5V DC $\pm 5\%$
EEPROM	4kByte
Temperaturbereich	-40 °C...+70 °C
Größe	12,5 x 12,5 x 5,2mm ³
Kompatibilität	Match-X; 0.63 (Pinbelegung siehe Datenblatt)
Kontaktierung	Top: Land Grid, (3 μm NiAu) Bottom – Variante 1: Land Grid (3 μm NiAu) Bottom – Variante 2: Steckkontakt (Steckermatrix)
Material	FR4
Farbmarkierung	AXY 2.1: Weiß/Weiß AXY2.2: Weiß / Schwarz

Abbildung 6.19: Datenblatt Funktionsblock Beschleunigungssensor der EasyKit Entwicklungsmethodik

6.4 Entwicklungs- und Testwerkzeuge

Für die Entwicklung von Funktionsträgern, Bausteinen und Systemen besteht in allen Stufen der Entwicklungsmethodik ein Bedarf für angepasste Entwicklungs- und Testwerkzeuge. Bei einer Eigenentwicklung stellen diese Testwerkzeuge

grundsätzlich einen zu berücksichtigenden Zeit- und Kostenfaktor dar. Aufgrund der einheitlichen Schnittstellen und Bauformen ist es jedoch möglich, im Rahmen der Entwicklungsmethodik spezifische Testsysteme zur Verfügung zu stellen, die wiederum von Dritten frei verwendet werden können.

Für die Entwicklung eigener Baugruppen (Stufe 1 und 2) der Entwicklungsmethodik besteht der Bedarf nach einem Adapter, der einen einfachen Zugriff auf den Funktionsträger ermöglicht und alle Anschlusspunkte in einem größeren Raster zur Verfügung stellt. Für diesen Zweck wurde ein Adapter entwickelt, der über das Anschlussraster der elektromechanischen Schnittstelle verfügt, eine entsprechende Öffnung für eine beidseitige Zugänglichkeit aufweist und gleichzeitig alle Anschlüsse des elektrischen Busses auf ein Raster von 2,54mm verteilt.

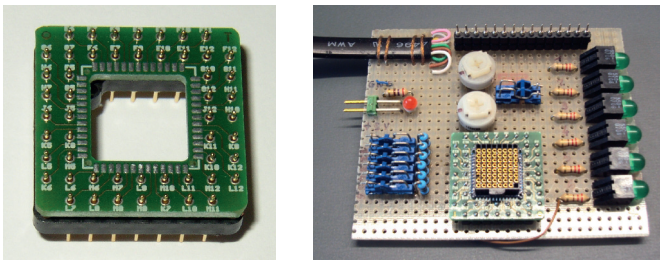


Abbildung 6.20: Umverdrahtungsadapter mit Sockel (links) und eingebunden in Testaufbau (rechts)

Dieser Adapter ermöglicht den Einsatz in Verbindung mit üblichen Rasterlochplatinen und lässt einen Zugriff auf die offen liegende Schaltung zu. In Verbindung mit entsprechenden Prüfstiften lassen sich so z.B. Signalpegel der Schaltung kontrollieren, auf die im geschlossenen Zustand eines Bausteins nicht mehr zugegriffen werden kann.

Darüber hinaus ermöglicht dieser Adapter auch, eine komplette Schaltung verschiedener Funktionsträger miteinander zu verdrahten und komplett offen zu betreiben. Da der Funktionsträger auf den Adapter aufgelötet werden muss, kann er jedoch nicht mehr in anderen Systemaufbauten genutzt werden.

Für teilweise oder komplett aufgebaute Bausteine, die bereits über die elektrischen Schnittstellen verfügen, wurde daher ein spezieller Testadapter (siehe Abbildung 6.21) entwickelt. Dieser verfügt über das Grid Array im Rastermaß von 1,27mm, das je nach Ausführungsform gleich mit einem Matrixstecker aufgebaut ist oder über die Federmatrix mit Bausteinen verbunden werden kann. Die

63 Anschlüsse des elektrischen Busses werden auf eine umlaufende doppelte Stiftleiste mit einem Rastermaß von 2,0mm geführt.

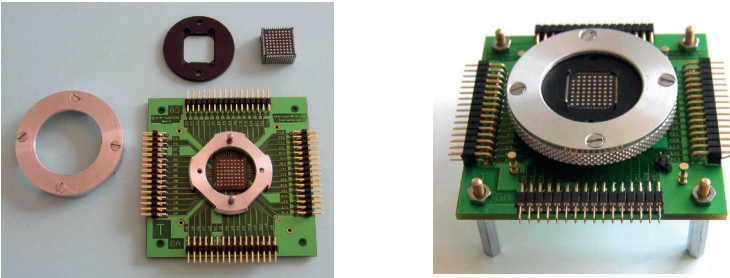


Abbildung 6.21: Testadapter (links) und mit eingesetztem Baustein (rechts)

Durch den Zugriff auf alle Anschlüsse besteht die Möglichkeit, einfach auf einzelne Signale zuzugreifen und diese sowohl für Testzwecke als auch für die Programmierung zu nutzen.

Für einen kompletten Systemtest (Stufe 2 der Entwicklungsmethodik) ist dieser Testadapter jedoch nur bedingt geeignet, da entweder mehrere Adapter aufwendig miteinander verdrahtet werden müssen oder beim Aufeinanderstecken mehrerer Bausteine die einzelnen Signalpfade zwischen den Bausteinen nicht mehr trennbar sind.

Daher wurde für den Systemtest ein komplexeres Testboard entwickelt (siehe Abbildung 6.22). Das Testboard verfügt wiederum zentral über eine Anschlussmöglichkeit für Bausteine, die wahlweise mit dem Matrixstecker oder der Federmatrix ausgeführt sein können. Alle 63 Anschlüsse des Bausteins sind nach außen geführt, hier jedoch auf beiden Stirnseiten des Testboards. Diese beiden Seiten verfügen über einen 96-poligen Stecker nach DIN41612, auf der Eingangsseite (links in Abbildung 6.22) als Stiftstecker und auf der Ausgangsseite (rechts) als Buchenstecker ausgeführt. Auf der Eingangsseite führen alle Signale über eine Leiste, die ein Trennen/Verbinden mittels Jumper ermöglicht und über eine Stiftleiste zur besseren Kontaktierung der Signale verfügt. Auf der Ausgangsseite ist ein weiterer 96-poligen Stiftstecker nach DIN41612 platziert. Dieser kann im einfachsten Fall mit einem entsprechenden Gegenstück überbrückt werden (direkte Durchleitung der Signale) oder über einen weiteren Adapter können die Signale von der Oberseite des Bausteins direkt auf diesen Stecker geführt werden.

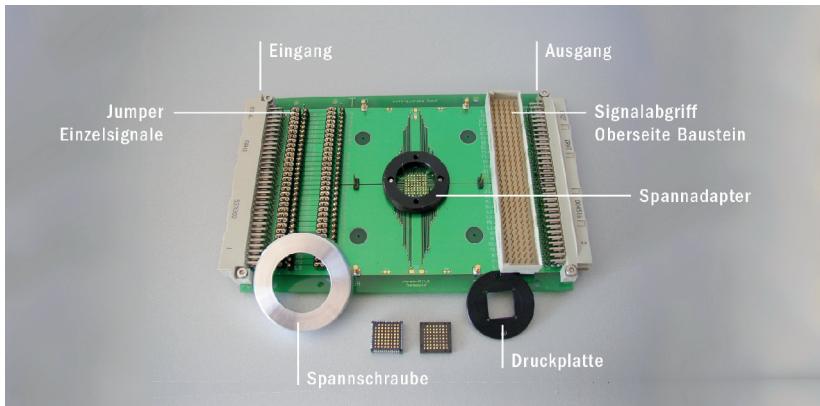


Abbildung 6.22: Testboard

Somit ist es möglich, die Signalführung eines einzelnen Bausteins so nachzubilden, dass sein Einsatz im Verbund direkt dargestellt und gleichzeitig jedes Signal (Ober- und Unterseite getrennt) zugänglich wird.

Werden nun mehrere dieser Testboards direkt miteinander verbunden, kann ein komplettes System in einer Ebene platziert werden, wobei alle Signale, insbesondere auch zwischen den Bausteinen, getrennt und untersucht werden können.

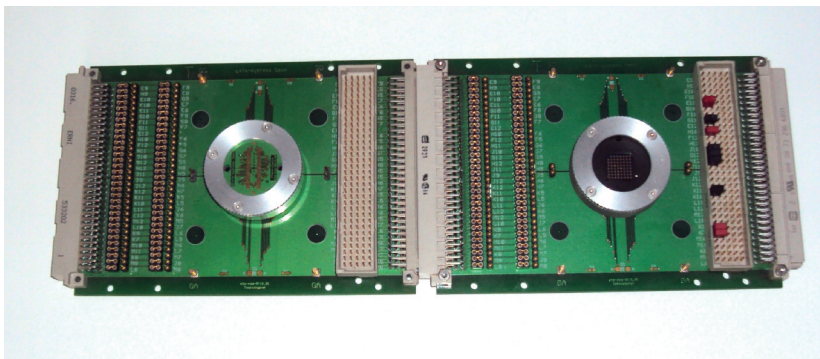


Abbildung 6.23: Systemanordnung mit 2 Bausteinen im Testboardverbund

Tabelle 6.1 gibt einen Überblick, welcher Adapter für welchen Einsatzzweck geeignet ist:

	Umverdrahtungsadapter	Testadapter	Testboard
Test von Funktionsträgern	sehr gut	Nicht möglich	Nicht möglich
Test von Bausteinen	Nur bei Bausteinen mit Matrixstecker möglich	sehr gut	Sehr gut, aber kostenintensiv
Programmierung von Bausteinen	Nur bei Bausteinen mit Matrixstecker möglich	sehr gut	Sehr gut, aber kostenintensiv
Test von Systemen	Sehr aufwendig	aufwendig	sehr gut

Tabelle 6.1: Vergleich der Eignung verschiedener Adapter für unterschiedliche Aufgaben

Durch die vorhandenen Testwerkzeuge wird somit der Gedanke des Rapid Prototyping auch bei der Entwicklung eigener Funktionsträger, Bausteine und Systeme unterstützt. In Verbindung mit den Layoutvorlagen und den Vorgaben zur Fertigung wird die Erstellung derartiger Funktionsblöcke effektiv und mit reduziertem Expertenwissen ermöglicht.

6.5 Programmierwerkzeug EasyLab

6.5.1 EasyLab – Die Idee

Für den industriellen Einsatz von Systemen mit integrierter Signalverarbeitung gelten hinsichtlich der zu erstellenden Software vielfältige Anforderungen, die sich je nach Einsatzfall stark unterscheiden können und von einer einfachen Signalwandlung bis hin zu einer komplexeren Signalverarbeitung mit Analysen und Buskommunikation reichen. Im einfachsten Fall erfolgen die Aufnahme eines Messwertes und die Übergabe dieses Wertes an ein Ausgabegerät über eine Kommunikationsschnittstelle. Die Verarbeitung der Messwerte kann eine einfache Wertberechnung (z.B. Skalierung) oder Zuweisung (z.B. look-up table) sein, sich aber je nach Anwendungsfall um eine Linearisierung, Korrektur und/oder Justierung erweitern. Hierfür sind meist noch Parameter zu ermitteln und in einem Speicher vor Ort abzulegen. Ein entsprechender Zugriff zum Ermitteln und Ablegen dieser Werte muss ermöglicht werden. Dieses Ermitteln und Ablegen führt sofort zur Fragestellung nach Zugriffsberechtigungen auf diese Werte. Im industriellen Umfeld sollte ein Zugriff sowohl auf das Programm als auch auf die Kalibrier- und Korrekturdaten nur autorisierten Personen möglich sein. Dies erfordert eine Verwaltung der Zugriffsrechte. Hier muss definiert werden, welcher Personenkreis Lese- und/oder Schreibrechte auf bestimmte Daten hat.

Analog zur Hardware soll auch hier der dreistufige Entwicklungsprozess begleitet und gleichzeitig eine möglichst hohe Wiederverwertbarkeit der erstellten Software in den einzelnen Stufen des Entwicklungskonzepts gewährleistet werden. Gleichzeitig soll durch die einfache Programmiermöglichkeit ein Zugang mit Grundkenntnissen der Programmierung möglich werden.

Hierfür wurde durch die TU München ein neues Werkzeug entwickelt, das grafische Programmiersystem EasyLab [BARNER 2008, GEISINGER 2009]. Dieses wurde speziell unter dem Gesichtspunkt erstellt, Nichtexperten im Bereich der Programmierung ein Werkzeug an die Hand zu geben, mit dem eigenständig einfache bis mittel komplexe Test- und Anwendungsprogramme selbstständig erstellt und auf das Zielsystem übertragen werden können.

Hierfür besteht das Programmiersystem aus mehreren Ebenen:

- Grafische Programmieroberfläche
- Funktionsbibliotheken
- Simulationsmodus
- Codegenerierung
- Debugging Modus
- Erweiterungsmodul

Diese werden im Weiteren vorgestellt.

6.5.2 Grafische Programmieroberfläche

Die grafische Oberfläche von EasyLab ermöglicht die Erstellung eines Software-Modells. Dieses Software Modell kann durch ein Datenflussprogramm und/oder eine Ablaufbeschreibung erstellt werden. Diese beiden Programmdarstellungen ergänzen sich und ermöglichen somit auch den Aufbau von komplexeren Programmen mit verschiedenen Zuständen. Beide Programmdarstellungen sind an die Norm IEC 61131-3 angelehnt, ohne diese vollständig umzusetzen.

Die Ablaufsprache beschreibt diese Zustände. Der Übergang in verschiedene Programzustände und die Bedingungen für diese Übergänge können über diese Oberfläche (in EasyLab auch als Hauptprogramm bezeichnet – siehe Abbildung 6.24) definiert werden.

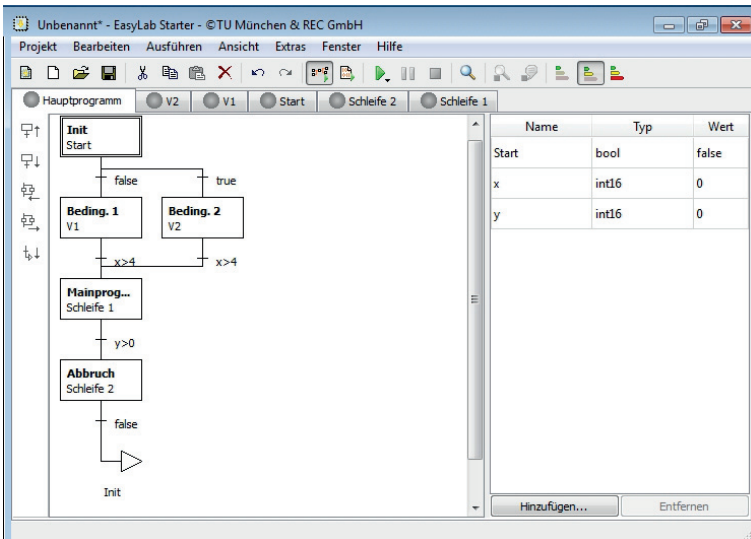


Abbildung 6.24: Grafische Programmieroberfläche von EasyLab – Hauptprogramm mit Ablaufschema

Jedes einzelne Rechteck des Hauptprogramms (Abbildung 6.24) beschreibt dabei einen Zustand. Jedem Zustand kann ein entsprechendes Unterprogramm zugeordnet sein. Solange eine bestimmte Bedingung, die als Ausgang des Zustandes definiert ist, nicht erfüllt wird, erfolgt die Ausführung dieses Zustandes bzw. seines Unterprogramms. Bei Erfüllung der Bedingung erfolgt der Übergang in den nächsten Zustand und dessen zugehöriges Unterprogramm. Das Ablaufprogramm ermöglicht neben sequenziellen Abläufen auch Verzweigungen oder Sprünge. Somit sind die Voraussetzungen gegeben, um Programme zu erstellen, die sich unter bestimmten eintretenden Umständen vollständig anders verhalten und in entsprechende Unterprogramme verzweigen oder springen können.

In den Unterprogrammen (Abbildung 6.25) wiederum wird auf die Datenflusssprache zurückgegriffen. Diese basiert auf Funktionsbibliotheken, die miteinander verknüpft werden können. Informationen und Daten werden in diesen Funktionen verarbeitet. Über Ein- und Ausgabeelemente können diese Daten in den Unterprogrammen eingelesen, verarbeitet und auch ausgegeben werden.

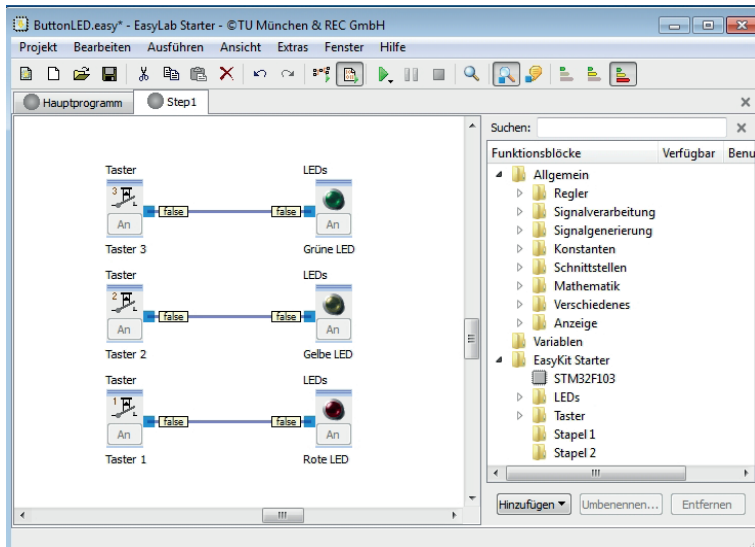


Abbildung 6.25: Grafische Programmieroberfläche von EasyLab – Unterprogramm mit Datenflussschema

6.5.3 Funktionsbibliotheken

In den Unterprogrammen stehen unterschiedlichste Funktionsbibliotheken zur Verfügung. Hierbei handelt es sich um Einheiten, die über Eingänge und/oder Ausgänge verfügen können und eigenständig Berechnungsoperationen durchführen. Für den Anwender ist nur das Wissen um die eigentliche Funktion der Bibliothek von Interesse (z.B. einer Multiplikation), nicht aber deren programmiertechnische Umsetzung.

Jedem Wert einer Funktionsbibliothek ist ein Datentyp zugeordnet. In EasyLab stehen die wesentlichen Datentypen Ganzzahlen, Kommazahlen und Wahrheitswerte zur Verfügung. Charakter sind in der dieser Arbeit zugrunde liegenden Version noch nicht umgesetzt. Eine Umwandlung von Datentypen ist prinzipiell möglich. Hierfür ist ein spezieller Funktionsblock zu verwenden.

In EasyLab wird in allgemeine, hardwareunabhängige und hardwarespezifische Funktionsblöcke unterschieden. Hardwareunabhängige Funktionsblöcke stehen grundsätzlich, unabhängig von der Zielplattform, zur Verfügung. Es handelt sich

hierbei um allgemeine Funktionen, z.B. mathematische Funktionen, Anzeigelemente, Reglertypen oder Filter. Die hardwareunabhängigen Funktionsblöcke sind somit auch auf allen Zielplattformen ausführbar. Bei der Zielplattform handelt es sich um die Hardware (Mikrocontroller mit Peripherie), auf der später das Programm ausgeführt werden soll. Im Gegensatz zu den hardwareunabhängigen Funktionen sind die hardwarespezifischen Funktionen nur in Verbindung mit der Zielplattform verwendbar. Es handelt sich hierbei z.B. um Ausgänge zur Ansteuerung von Komponenten, um hardwareabhängige Eingänge oder um Funktionen für die Kommunikation.

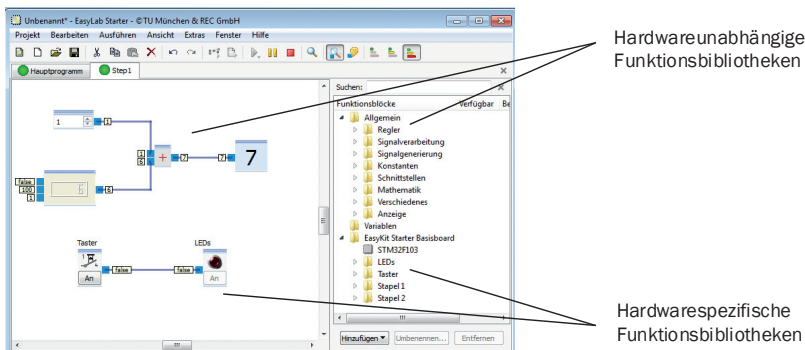


Abbildung 6.26: Hardwareunabhängige und hardwarespezifische Funktionsbibliotheken

In den hardwarespezifischen Funktionsbibliotheken kann bereits eine umfassendere Signalvorverarbeitung stattfinden. So liefert der z.B. die Funktionsbibliothek für den analogen Spannungseingang 0...5V DC neben den A/D-Wandlerrohwerten auch direkt die Spannung.

Nach erfolgter Programmerstellung kann dieses direkt in EasyLab ausgeführt werden. Hierfür stehen zwei unterschiedliche Modi zur Verfügung, der Simulationsmodus und der Modus Codegenerierung.

6.5.4 Programmausführung

Programmausführung – Simulation

Der Simulationsmodus dient der Prüfung des erstellten Programms, unabhängig von der Zielplattform. Im Simulationsmodus erfolgt der Programmablauf aus-

schließlich auf dem Rechner, auf dem das EasyLab-Programm installiert ist. Eingangs- und Ausgangswerte können virtuell vorgegeben und während der Programmausführung verändert werden. Ebenso ist es zum Teil möglich, hardware-spezifische Komponenten und deren Verhalten nachzubilden.

Der Simulationsmodus ist somit besonders in der Entwicklungsphase hilfreich, um die erforderlichen Softwaresystemkomponenten zu ermitteln, das Programm vorab zu testen und auch den Bedarf an die Zielplattform, insbesondere die erforderliche Leistungsfähigkeit des Controllers zu ermitteln.

Programmausführung - Codegenerierung

Die zweite Möglichkeit in EasyLab ist die direkte Erzeugung des Programmcodes verbunden mit der Übertragung des Programmcodes auf die Zielplattform. Dafür müssen verschiedene Voraussetzungen erfüllt sein. Auf der Zielplattform muss ein Bootloader installiert sein, der über eine Kommunikationsschnittstelle eine Übertragung des erzeugten Programmcodes ermöglicht. Die Kommunikationsschnittstelle kann dabei z.B. eine USB-Schnittstelle oder eine Feldbusschnittstelle (RS485, CAN) sein.

In EasyLab ist es erforderlich, effizienten und robusten Programmcode für die jeweilige Zielplattform zu erzeugen. Dafür müssen die allgemeinen, plattformunabhängigen Funktionen schrittweise in ein plattformabhängiges Modell umgewandelt werden. Aus diesem Modell wird wiederum der Code für die Zielarchitektur erzeugt. Die Programmiersprache des Codes ist C. Vorteile hierbei sind, dass bei den meisten Zielplattformen C-Compiler verfügbar sind und gleichzeitig C einen guten Kompromiss bei der Effizienz des erzeugten Codes bietet.

Die erforderlichen Compiler werden je nach Zielplattform gleich mit der Installation von EasyLab mitgeliefert und installiert oder sind zusätzlich zu erwerben und gesondert zu installieren. Im Menü Codegenerierung der EasyLab Software (siehe Abbildung 6.27) können alle erforderlichen Einstellung zur Zielplattform, zur Werkzeugkette und für den Bootloader vorgenommen werden.

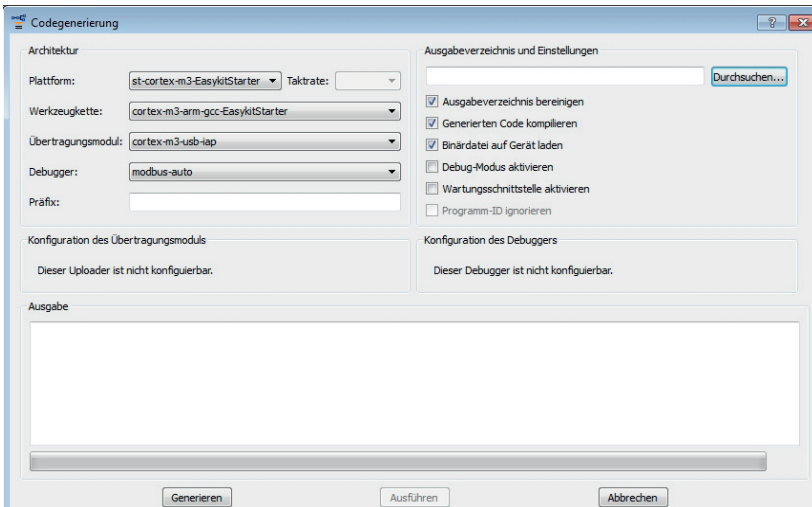


Abbildung 6.27: Menü Codegenerierung

Der so erzeugte Code wird dann direkt auf die Zielplattform übertragen und dort ausgeführt. Auch dabei können noch einmal zwei verschiedene Modi gewählt werden, mit oder ohne Debugging Funktionalität.

6.5.5 Interaktives Debugging

EasyLab bietet die Möglichkeit eines interaktiven Debuggings. Hierzu ist der Programmcode im entsprechenden Modus auf die Zielplattform zu übertragen. Während der anschließenden Ausführung des Codes auf der Zielplattform bleibt diese mit dem Rechner verbunden und überträgt die Werte aller Eingänge und Ausgänge der Funktionsbibliotheken an diese. Somit kann die Programmausführung und damit verbunden der komplette Datenfluss überprüft und ggf. korrigiert werden. Darüber hinaus können auch bestimmte Eigenschaften (z.B. Werte von Konstanten und Variablen) im laufenden Betrieb angepasst werden. Diese Werte werden direkt auf die Zielplattform übertragen und dort im laufenden Betrieb geändert.

Das interaktive Debugging bietet somit eine effektive und effiziente Möglichkeit, das erstellte Programm zu testen, anhand des Datenflusses und der angezeigten Werte Fehler zu finden sowie das Programm zu optimieren. Darüber hinaus

bietet dieser Modus auch die Möglichkeit, bereits installierte Geräte (z.B. bei Wartungsaufgaben) in Verbindung mit der realen Einsatzumgebung zu prüfen und so den Service zu vereinfachen.

Die Ausführung des Programms im Debugging Mode ist mit entsprechendem zusätzlichem Code verbunden. Bei Mikrocontrollern mit geringen Ressourcen (Speicherplatz) kann dies bereits in der Entwicklungsphase dazu führen, dass der Code nicht mehr im Debugging Modus ausführbar ist.

Für den regulären Betrieb erfolgt die Codegenerierung ohne die Debugging Funktionalität. Das System ist somit eigenständig lauffähig und enthält nur den funktionsrelevanten erstellten Code.

6.5.6 Erweiterungsmodul

Für Anwender des Systems ist eine Erweiterung in zwei verschiedenen Bereichen von großer Bedeutung, der Einbindung neuer Funktionsbibliotheken sowie der Einbindung neuer Zielplattformen und damit verbunden der dafür erforderlichen Werkzeugketten.

Die einfache Erweiterbarkeit der Funktionsbibliotheken ist ein zentraler Punkt. Wie bei allen modularen Systemen ist davon auszugehen, dass je nach Ausbauzustand nur ein Teil der tatsächlich erforderlichen Funktionen zur Verfügung stehen. Dem Anwender muss somit die Möglichkeit eröffnet werden, eigene Funktionen, Geräte oder Laufzeitumgebungen zu erstellen und dem System hinzuzufügen. Die Vorgehensweise hierzu ist im Handbuch EasyLab-Dokumentation für den Entwickler [EasyLab2012] beschrieben.

Bei der Einbindung einer neuen Zielplattform ist die komplette Werkzeugkette zu berücksichtigen. Diese reicht vom Compiler und Linker über die Dateikonvertierung bis zur Übertragung auf das Zielsystem. Diese Anpassungen erfordern in der Regel umfassendes Expertenwissen und können demzufolge nur von entsprechenden Spezialisten durchgeführt werden.

7 Anwendungsbeispiele

Die Entwicklungsmethodik EasyKit soll die Entwicklung von Prototypen beschleunigen (Rapid Prototyping) und gleichzeitig auch einen schnelleren Übergang dieser in ein serienreifes Produkt sicherstellen [BAUER 2009A, BÖNICKE 2010]. Zur Unterstützung dieser These soll an zwei Beispielen – einem Temperaturremesssystem mit einem Pt100 Sensor und einem kombinierten Sensor-/Aktorsystem in Form eines positionsgesteuerten Pneumatikzylinders – der Aufbau von Systemen basierend auf der Entwicklungsmethodik EasyKit dargestellt werden. Am Beispiel des Messsystems wird die Weiterentwicklung bis zu einem serienreifen Produkt (Stufe 3) aufgezeigt. Gleichzeitig werden bei diesem Beispiel auch das konventionelle Vorgehen bei der Programmierung und die Programmerstellung mittels EasyLab verglichen, um so im Ergebnis einen Vergleich der Wirtschaftlichkeit durchführen zu können.

7.1 Temperaturremesssystem Pt100

Die folgende Beschreibung orientiert sich am angepassten Entwurfsalgorithmus für die Entwicklungsmethodik EasyKit unter Berücksichtigung des V-Models für die Entwicklungsmethodik EasyKit (siehe Abbildung 4.3 und Abbildung 4.4). Dabei ist zu berücksichtigen, dass der Makrozyklus in jeder Entwicklungsstufe 1-3 durchlaufen wird, Anpassungen der Anforderungen jedoch nicht vorgenommen werden.

7.1.1 Allgemeine Anforderungen

Zielstellung ist die Entwicklung eines Temperaturremesssystems mit folgenden grundlegenden Parametern:

- Erfassung und Wandlung der Messwerte eines Pt100 in einem Messbereich von 0 °C bis 250 °C
- Auflösung Messwertanzeige: 0,01 °C
- Linearisierung und Berechnung des Temperaturwertes anhand der Callendar-van-Dusen Gleichung [Irrgang 2004]
- Anschluss eines Pt100 in Vierleiterschaltung
- Übertragung des Messwertes mittels RS485/Modbus
- Möglichkeit des Abgleichs (Justierung) der Messkette

- Bauform: Rundgehäuse ø 20mm
- Steckeranschluss Kommunikation und Versorgung: M12, 4-polig

7.1.2 Funktionsstruktur und Teilfunktionen

In einem ersten Schritt erfolgt, auch unabhängig von der einzusetzenden Entwicklungsmethodik, die Zerlegung des geplanten Gesamtsystems in seine Teilfunktionen. Dabei ist die Zielstellung zu berücksichtigen, die komplette Messkette in einem System abzubilden. Hierfür ist von folgenden grundlegenden Funktionen auszugehen:

Funktion	Beschreibung
Spannungsversorgung/ -adaption	Anpassung der anliegenden externen Spannung (12VDC oder 24VDC) an die noch zu definierende systeminterne Spannung
Kommunikation	Bereitstellung der hardwaretechnischen Voraussetzungen für eine RS485 Kommunikation
Anschluss Spannung und Kommunikation	Verbindungsmöglichkeit zu externen Komponenten über Sensorstecker M12
Signalverarbeitung	Berechnung des Temperaturwertes aus dem ADC-Rohwert, Korrektur des Signals entsprechend Forderungen Bereitstellung der softwaretechnischen Voraussetzungen für die Kommunikation
Signalwandlung	Umwandlung der temperaturabhängigen Änderung des Widerstandes des Pt100 und der daraus resultierenden Spannungsänderung in einen Digitalwert
Anschluss Pt100	Anschluss in 4 Leiterschaltung, Bereitstellung eines entsprechenden Verbindungselementes

Tabelle 7.1: Aufstellung der Teilfunktionen

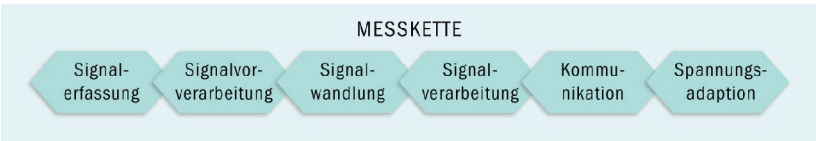


Abbildung 7.1: Funktionen des Temperaturmesssystems

7.1.3 Aufbau des Systems nach Stufe 1 der Entwicklungsmethodik

In einem weiteren Schritt erfolgen die detaillierte Beschreibung der Einzelfunktionen und die Überprüfung auf vorhandene Funktionsblöcke in der Entwicklungsmethodik EasyKit.

Signalvorverarbeitung und Signalwandlung

Aus den oben beschriebenen Anforderungen ergibt sich ein Messbereich von 250K bzw. eine Anforderung an eine Auflösung von 25.000 Digits d.h. mindestens 15Bit. Die EasyKit Entwicklungsmethodik verfügt über einen Baustein mit einem direktem A/D-Wandlereingang (24bit). Dieser hat einen Messbereich von -2,5V bis + 2,5V. Für diesen Messbereich ergibt sich somit bei einer Berücksichtigung der 24bit eine theoretisch maximale Auflösung von 298nV.

Gleichzeitig ermöglicht der Baustein einen Anschluss des Pt100 in 4-Leiterschaltung. Hierfür verfügt der Baustein über eine optionale Bestückmöglichkeit im Versorgungszweig des Pt100. Bei Verwendung von 10k Vorwiderständen wird der Pt100 von einem Strom von 210 μ A durchflossen (siehe Abbildung 7.2). Eine Änderung von 0,01K führt somit zu einer Widerstandsänderung von 3,9mOhm bzw. zu einer Spannungsänderung über den Pt100 von ca. 1 μ V. Die geforderte Auflösung kann demzufolge durch Einsatz dieses Bausteins ohne eine zusätzliche Signalverstärkung erreicht werden.

Auf Basis dieser Prüfung erfolgte die Auswahl des Bausteins mit der Bezeichnung ADC2.0 für die Teilfunktionen Signalvorverarbeitung und Signalwandlung.

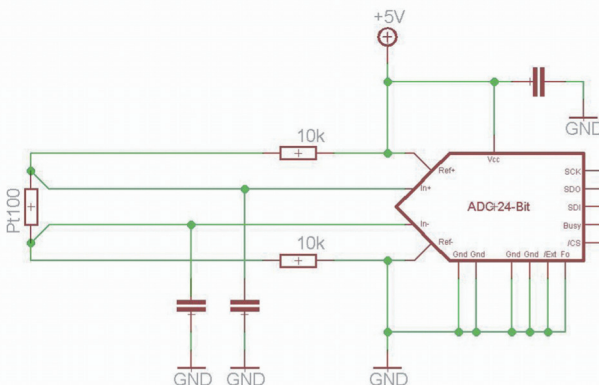


Abbildung 7.2: Eingangsbeschaltung des ADC-Bausteins mit 24bit AD-Wandler

Signalverarbeitung und Kommunikation

Für die Signalverarbeitung und Kommunikation stehen in EasyKit CPU-Bausteine zur Verfügung, die beide Funktionen direkt miteinander verbinden. Die Entwicklungsmethodik EasyKit bietet nun wie bereits beschrieben die Möglichkeit des frühzeitigen Hardware/Software-Codesigns, d.h. bereits in dieser Phase des Entwurfsprozesses erfolgt ein Abgleich zwischen zur Verfügung stehender Hardware und deren Ressourcen (hier in Form des CPU-Bausteins) und dem Bedarf der Software (hier in Form der Berechnungsroutinen und der Buskommunikation – Beschreibung des parallelen Vorgehens in Kapitel 7.1.4).

Im vorliegenden Beispiel sind folgende grundlegende Operationen erforderlich:

- Wandlung des ADC-Rohwertes in den äquivalenten Wert der Eingangsspannung
- Umrechnung der Spannung in den Temperaturwert (Calendar-van-Dusen Gleichung)
- Mittelwertbildung
- Verschiebung der Kennlinie an bekannten Punkten (Justierung)
- Kommunikation mittels Modbus RTU

Auf Basis dieser Prüfung erfolgt die Auswahl des Bausteins CPU2.2 für die Signalverarbeitung und Kommunikation. Dieser Baustein verfügt über einen 8bit Mikrocontroller (PIC18F2520) mit externem Quarz sowie einem integrierten RS485 Treiber.

Spannungsadaption

Die Anforderungen an die Spannungsadaption sind die Transformation einer externen Eingangsspannung von 12VDC oder 24VDC auf die interne Systemspannung. Die ausgewählten Bausteine ADC2.0 und CPU2.2 arbeiten mit 5VDC. Hierbei ist zu berücksichtigen, dass der Baustein ADC 2.0 über zwei getrennte 5V Anschlüsse verfügt, sowohl für die Versorgung der digitalen Bauelemente als auch für die analogen Komponenten.

Für diese Anforderungen steht der Baustein SRM2.0 zur Verfügung. Dieser verfügt über einen Eingang mit Verpolungs- und Überlastschutz und zwei getrennten Ausgängen (analog und digital), die je nach Bedarf (Bestückungsoption) mit 3,3V oder mit 5V ausgeführt werden können. In Verbindung mit einer Interface-

platine kann der Baustein so komplettiert werden, dass ein direkter Anschluss über Stecker an externe Komponenten ermöglicht wird. Somit können alle erforderlichen Teilfunktionen des Gesamtsystems mit Bausteinen aus dem EasyKit Baukasten erfüllt werden, ein direkter Systemaufbau ist möglich.

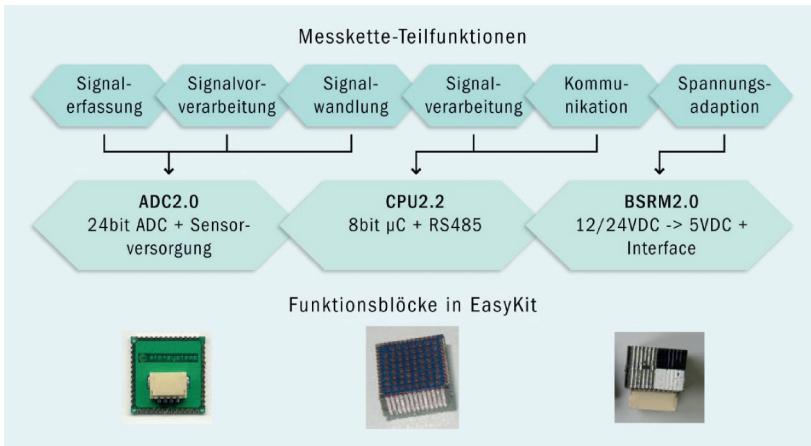


Abbildung 7.3: Zuweisung der EasyKit Funktionsblöcke zu den erforderlichen Teilfunktionen

Für ein erstes Funktionsmuster werden die Bausteine über die Matrixstecker miteinander verbunden. Ein Pt100-Fühler wird mit dem erforderlichen Anschlussstecker (JST-SR04) konfektioniert und kann somit direkt auf den ADC-Baustein aufgesteckt werden. An den BSRM2.0 Baustein wiederum kann direkt ein Sensorstecker M12/4-polig angeschlossen werden, der ebenfalls in EasyKit als Funktionsblock und -baustein vorliegt.

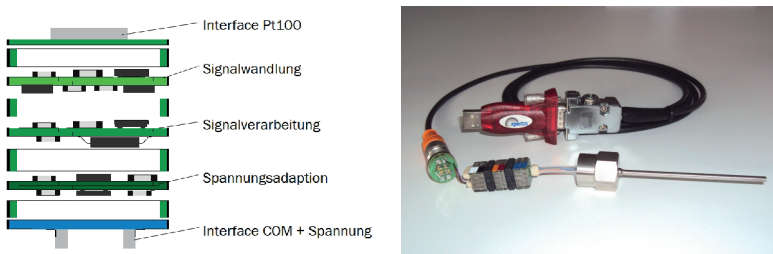


Abbildung 7.4: Pt100 Messsystem aus EasyKit Bausteinen mit Anschlussset (links: Prinzipaufbau, rechts: realisiertes System)

Der Systemverbund wird über den Anschlussstecker M12/RS485 und ein Verbindungskabel sowie einen RS485/USB-Konverter direkt an den PC angeschlossen. Die Versorgung des Systems erfolgt über ein 24V Steckernetzteil.

7.1.4 Vorgehen zur Ermittlung der Temperatur aus dem A/D-Wandlersignal

Parallel zur Auswahl des CPU-Bausteins erfolgen die Erstellung des Softwaremodells und die Prüfung der zur Verfügung stehenden Hardwareressourcen.

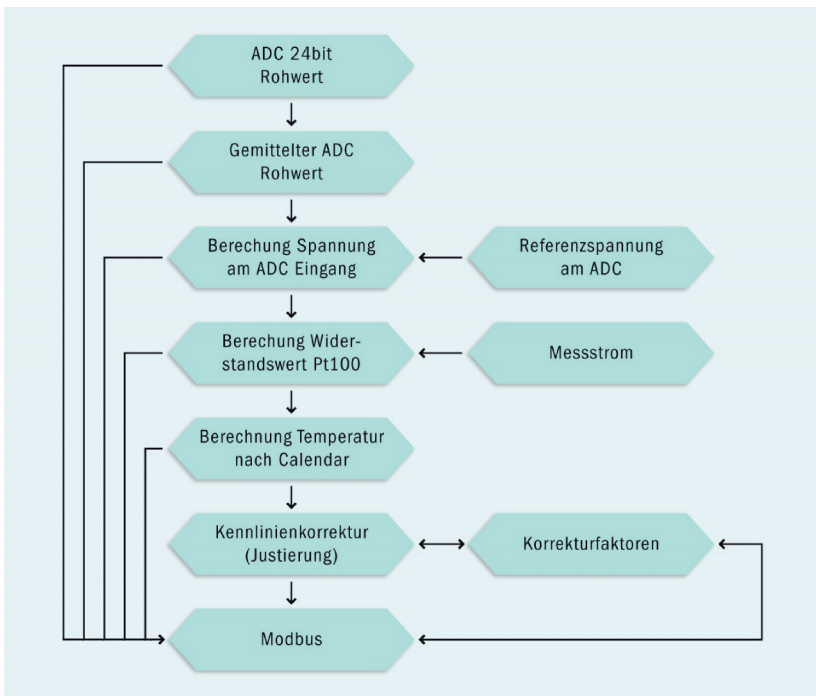


Abbildung 7.5: Ablaufschema der Firmware für das Temperaturmesssystem

Über den SPI-Bus sind die Rohwerte des eingesetzten ADC abzuholen. In einem ersten Schritt erfolgt die Mittelung der Rohwerte. Diese Art der Mittelung ist freigestellt, der Mittelwert sollte über 2 bis max. 16 Werte gebildet werden.

Anschließend ist aus dem ADC Wert die am ADC-Eingang anliegenden Spannung zu berechnen. Hierfür muss die am Referenzeingang des ADC anliegende Spannung berücksichtigt werden. In einem folgenden Schritt ist der aktuell gemessene Widerstandswert des Pt100 unter Berücksichtigung des Messstroms zu berechnen.

Mit diesem Wert ist die Grundlage für die folgenden Temperaturwertberechnungen geschaffen. Gemäß Forderung ist die Calendar-van-Dusen Gleichung zu verwenden [Irrgang 2004]:

$$R(T) = R_0(1 + AT + BT^2 + C(T - 100^\circ\text{C})T^3) \quad (7-1)$$

mit den Koeffizienten

$$A = +3,9083 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1}$$

$$B = -5,7750 \cdot 10^{-7} \text{ } ^\circ\text{C}^{-2}$$

$$C = 0 \text{ für } 0 < T[^\circ\text{C}] < 850$$

$$R_0 = 100 \text{ } \Omega$$

Diese Gleichung lässt sich unter Vernachlässigung des Koeffizienten C ($T > 0^\circ\text{C}$) wie folgt für die Berechnung der Temperatur umstellen:

$$T = \frac{-R_0 * A + \sqrt{R_0^2 * A^2 - 4 * R_0 * B * (R_0 - R)}}{2 * R_0 * B} \quad (7-2)$$

mit

T – gesuchter Temperatur

R – gemessener Widerstandswert

R_0, A, B – Koeffizienten nach Gleichung 6-1

In einem weiteren Schritt soll die Möglichkeit bestehen, den so berechneten Wert noch zu korrigieren (Offset und Neigung), idealerweise durch Ermittlung des realen Temperaturwertes in einem Kalibrierbad und selbständiger Berechnung der Korrekturfaktoren bei Eingabe des realen Temperaturwertes.

Alle berechneten Werte sind über Modbus zur Verfügung zu stellen. Die Korrekturwerte wie auch die Werte der Spannungsreferenz und des Messstromes können über Modbus dem System mitgeteilt werden.

Für die Auswertung am PC steht ein Modbus Programm inkl. einer Modbus DLL zur Verfügung, mittels der die Registerwerte ausgelesen, angezeigt und bei Bedarf (Justierung) auch vom Master zum Slave übertragen werden können.

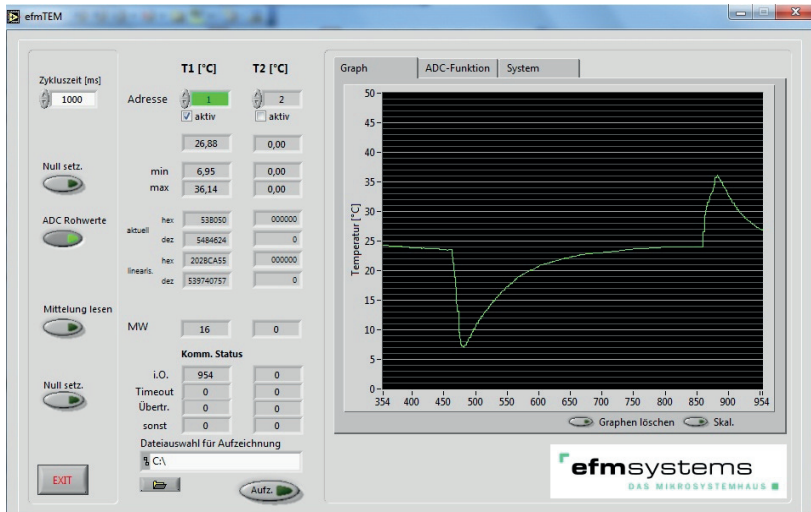


Abbildung 7.6: Visualisierungssoftware

Bis zu diesem Punkt ist das Softwaremodell unabhängig von der Entwicklungsmethodik. Die Vorteile werden nun erst durch einen Vergleich mit dem konventionellen Vorgehen ersichtlich.

7.1.5 Erstellung der Firmware mittels konventionellem Ansatz

Die Programmerstellung für das Temperatormesssystem erfolgt hierbei mittels C. Das Programm wird prinzipiell in drei Hierarchieebenen aufgeteilt, dem Bootloader, der Kommunikation und der Datenverarbeitung. Zusätzlich wird ein Bereich zur Verwaltung unterschiedlicher Datensätze vorgesehen.

Um einen Programmaustausch (Flashen) im zusammengebauten Zustand zu ermöglichen, ist ein Bootloader erforderlich. Dieser ermöglicht direkt über die Feldbusschnittstelle (hier RS485/Modbus) eine Übertragung einer neuen Firmware. Nach jedem Neustart durchläuft das Programm den Bootloader-Modus. Erfolgt eine direkte Anfrage, kann das Programm eine Verbindung aufbauen und

die Einspielung eines neuen Codes ist möglich. Der Zeitaufwand für die Neuerstellung dieses Codeteils ist mit ca. 8 AT auf den vorgeschlagenen Mikroprozessor erforderlich sein, ist mit einem Aufwand von ca. 3-4 AT auszugehen.

Als Kommunikationsprotokoll für die RS485-Schnittstelle wurde das Modbus-Protokoll [Modbus 2006] definiert. Es basiert auf einer Master/Slave-Architektur. Da mit einem Master mehrere Slaves verwaltet werden können, bietet Modbus eine einfache Möglichkeit, komplexere Netzwerke aufzubauen. Für den Einsatzfall wurde das Modbus RTU Protokoll mit einer Baudrate von 19.200 definiert. Der Messwert ist über ein Holding Register zu übertragen. Idealerweise werden darüber hinaus wie gefordert auch alle Zwischenwerte (ADC-Rohwert, Spannungswert, Widerstand, nicht korrigierter Temperaturwert, korrigierter Temperaturwert) über entsprechende Register übertragen. Der Zeitaufwand für die Neuerstellung dieses Codeteils ist ebenfalls mit ca. 8 AT zu planen, bei Wiederverwendung eines vorhandenen Codes, der lediglich auf den Prozessor anzupassen ist, kann ebenfalls von ca. 3-4 AT ausgegangen werden.

Die Ebene der Datenverarbeitung enthält den Programmteil zur Messwertverarbeitung und -aufbereitung. Hier ist der anwendungsspezifische Teil der Firmware zu implementieren. Erstellung und Test dieses Teils werden mit ca. 5 AT veranschlagt. Hier kann davon ausgegangen werden, dass auf keinen vorhandenen Code zurückgegriffen werden. Aufgrund der begrenzten Möglichkeiten des verwendeten Mikrocontrollers PIC18F2520 sind die meisten arithmetischen Funktionen selbst zu programmieren.

Durch den Aufbau in einzelnen Funktionsbausteinen ist zum Teil auch softwaretechnische Funktionalität an diese ausgelagert. So verfügt der ADC Baustein über einen eigenen Speicher, in dem bausteinspezifische Daten ausgelagert werden können. In diesem konkreten Fall sind dies die spezifischen Justierdaten sowie die Erkennungsdaten des Bausteins (Typ, Hersteller). Der Aufwand für Erstellung des Programmcodes für die Auslagerung und den Zugriff auf diese Daten beträgt max. 1 AT.

Somit ist je nach Ausgangslage (Verfügbarkeit von C-Bibliotheken) mit einem Zeitaufwand von 12 bis 20 Arbeitstagen zu rechnen. Ein frühes Hardware/Software-Codesign ist mit diesem Vorgehen nur bedingt möglich. Eine genaue Überprüfung der Eignung des ausgewählten Prozessors führt zu einer entsprechenden Wartezeit bei der Weiterentwicklung der Hardware.

Im konkreten Beispiel ist nach Erstellung und Übertragung aller Programmteile der Speicherplatz des Controllers zu ca. 85% ausgelastet.

Die mittels konventionellen Vorgehen erstellte Firmware wurde mittels eines Simulators (SIKA2000+) getestet. Mit diesem Gerät können Pt100 Signale simuliert und dem Messsystem in einem Bereich von -100°C ... $+600^{\circ}\text{C}$ mit einer Auflösung von 0,01K vorgegeben werden. Mittels einer Visualisierungssoftware (siehe Abbildung 7.6) können die Ergebnisse angezeigt und bei Bedarf auch abgespeichert werden.

Für den Test wurden in einem ersten Schritt Werte zwischen 0°C und 250°C angefahren und geprüft. Anschließend fand eine Justierung an zwei Punkten (0°C und 100°C) statt. Hierzu wurden diese beiden Punkte am SIKA eingestellt und dem Messsystem jeweils der eingestellte Punkt übergeben, so dass der interne Korrekturalgorithmus arbeiten kann. Anschließend wurden die zuvor überprüften Messpunkte erneut angefahren und in dieser Einstellung einer Prüfung unterzogen. Die Ergebnisse sind in Tabelle 7.2 dargestellt.

Vorgabewert	Messwert ohne Justierung	Messwert mit Justierung
0°C	$0,57^{\circ}\text{C}$	$0,00^{\circ}\text{C}$
30°C	$30,55^{\circ}\text{C}$	$29,98^{\circ}\text{C}$
60°C	$60,53^{\circ}\text{C}$	$59,99^{\circ}\text{C}$
90°C	$90,50^{\circ}\text{C}$	$89,99^{\circ}\text{C}$
120°C	$120,48^{\circ}\text{C}$	$119,98^{\circ}\text{C}$

Tabelle 7.2: Genauigkeit des Systems mit konventionell erstellter Firmware

Im Ergebnis ist zu sehen, wie durch die Justierung des Systems an zwei Punkten die Genauigkeit deutlich gesteigert werden kann.

Zusammenfassend ist festzustellen, dass der Aufwand für die Erstellung und den Test der Firmware unter Berücksichtigung bereits vorhandener Programmteile (Bootloader und Modbus Kommunikation) ca. 12...20 AT beträgt. Die geforderten Eigenschaften des Systems können dabei zu 100% erfüllt werden.

7.1.6 Erstellung der Firmware mittels EasyLab

Für die Erstellung einer Firmware mittels EasyLab ist das Vorhandensein eines Bootloaders auf der zu programmierenden Zielplattform zwingend erforderlich. Für die Zielplattform, den Baustein CPU2.2 liegt dieser Bootloader in EasyLab bereits vor.

Für die Modbus Kommunikation steht in EasyLab ebenfalls ein Funktionsblock zur Verfügung (siehe Abbildung 7.7). In diesen können Werte geschrieben, der Datentyp des Wertes festgelegt sowie die Registernummer definiert werden. Des Weiteren kann definiert werden, ob der Wert vom Master aus veränderbar ist. Damit erfüllt diese Modbusfunktion alle Eigenschaften, die laut Aufgabenstellung erforderlich sind. Es ist jedoch anzumerken, dass nur für Holdingregister eine Funktion vorliegt, nicht für die anderen Typen (z.B. Inputregister).

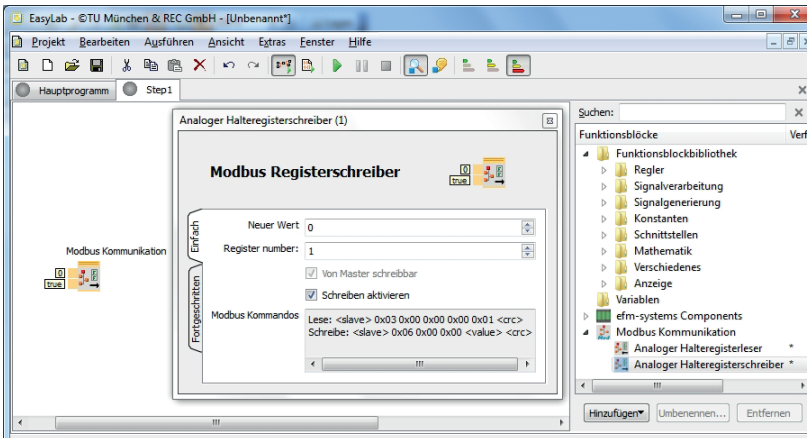


Abbildung 7.7: Modbus Funktion und Detailbeschreibung

Somit kann nun die Erstellung der eigentlichen, anwendungsspezifischen Datenerfassung in EasyLab erfolgen. Für den Baustein ADC2.0 steht ein eigener Funktionsblock zur Verfügung, der den Digitalwert des ADC als 32bit Rohwert zur Verfügung stellt. Dieser ist in einem ersten Schritt zu skalieren und in die am Eingang liegende Spannung umzurechnen. Anschließend wird unter Berücksichtigung des Messstroms der Widerstandswert des Pt100 ermittelt.

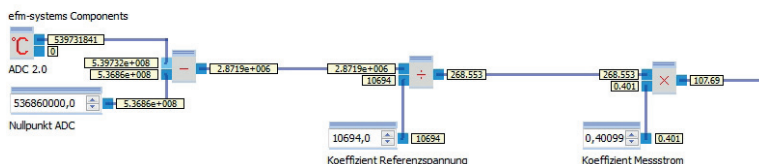


Abbildung 7.8: Umrechnung des ADC Rohwertes in die Spannung und Widerstand über den Pt100

Durch die Debugging Funktion von EasyLab können diese Schritte nachvollzogen und die Werte an jedem Funktionsblock auf Plausibilität überprüft werden.

Der so ermittelte Widerstandswert kann durch Verwendung von Gleichung 6-2 direkt in die Temperatur überführt werden. Zu beachten ist hierbei allerdings, dass in EasyLab derzeit keine Funktion für die Quadratwurzel oder eine vergleichbar einsetzbare Funktion zur Verfügung steht. Daher muss diese Funktion nachgebildet werden.

Dabei kann z.B. durch das Heron Verfahren mittels einfacher Iteration eine gute Annäherung erreicht werden. Dabei ist aus der Zahl a die Quadratwurzel zu ermitteln. Es gilt:

$$X_{n+1} = \frac{X_n + \frac{a}{X_n}}{2} \quad (7-3)$$

X_0 = Startwert

n – Anzahl Iterationen

Der Startwert X_0 kann beliebig festgelegt werden.

Für eine erste Programmversion wurde von drei Iterationsschleifen ausgegangen. Nach Abbildung der Gleichung 6-2 unter Verwendung des Heron-Verfahrens (6-3) liegt somit bereits der berechnete Temperaturwert vor.

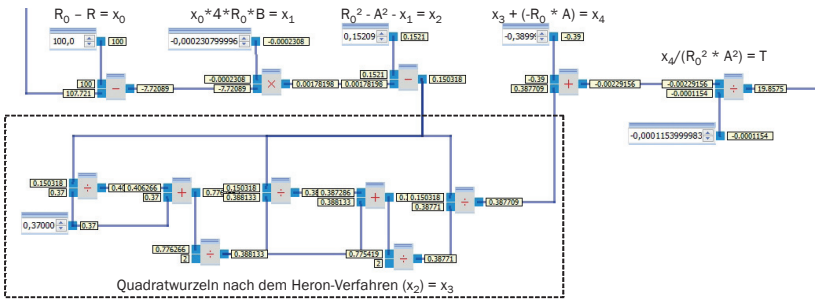


Abbildung 7.9: EasyLab Programm für Temperatursensorsystem

Im abschließenden Schritt erfolgt die Korrektur des Messwertes mittels:

$$T_{korr} = AT - B \tag{7-4}$$

mit

T_{korr} – Korrigierter Temperaturwert

A – Steigung

B – Offsetverschiebung

Für den Test wurden die gleichen Werte wie bei den Tests der konventionellen Firmware-Variante zwischen 0 °C und 120 °C angefahren und geprüft. Anschließend fand eine Justierung an zwei Punkten (0 °C und 100 °C) statt. Hierzu wurden diese Punkte am SIKA eingestellt. Die Korrekturfaktoren für die Justierung wurden manuell ermittelt und in die Software eingetragen (siehe Abbildung 7.10). Anschließend wurden die zuvor überprüften Messpunkte erneut angefahren und in dieser Einstellung einer Prüfung unterzogen. Die Ergebnisse sind in Tabelle 7.3 dargestellt.

Vorgabewert	Messwert ohne Justierung	Messwert mit Justierung
0 °C	0,30 °C	0,00 °C
30 °C	30,15 °C	29,93 °C
60 °C	59,90 °C	59,99 °C
90 °C	89,65 °C	90,03 °C
120 °C	119,38 °C	119,98 °C

Tabelle 7.3: Genauigkeit des Systems mit EasyLab Firmware

Die Erstellung der Firmware für die Datenverarbeitung des Temperaturmesssystems konnte in EasyLab von einem Maschinenbauingenieur ohne Erfahrung in der Programmierung in einem Zeitrahmen von ca. 5h realisiert werden.

In einem abschließenden Schritt war die Datenverarbeitung mit der Modbus Kommunikation zu kombinieren. Dies war jedoch mit der vorliegenden EasyLab Version nicht mehr im Debugging Modus möglich, da die Speicherressourcen des verwendeten Mikrocontrollers überschritten wurden (Info erfolgt durch Fehlermeldung mit Hinweis im Programm). Bei einem Wechsel in den normalen Codeerzeugungsmodus ohne Debugging konnte das Programm (siehe Abbildung 7.10) wiederum ausgeführt, auf die Zielformat übertragen und dort erfolgreich ausgeführt werden. Dies bedeutet gleichzeitig, dass die Debugging Funktionalität ca. 30% der Ressourcen dieses Mikrocontrollers in Anspruch nimmt. Eine Auslagerung von Funktionen zur Datenverwaltung an den EEPROM des ADC-Bausteins ist mit EasyLab in der aktuellen Ausbaustufe nicht möglich.

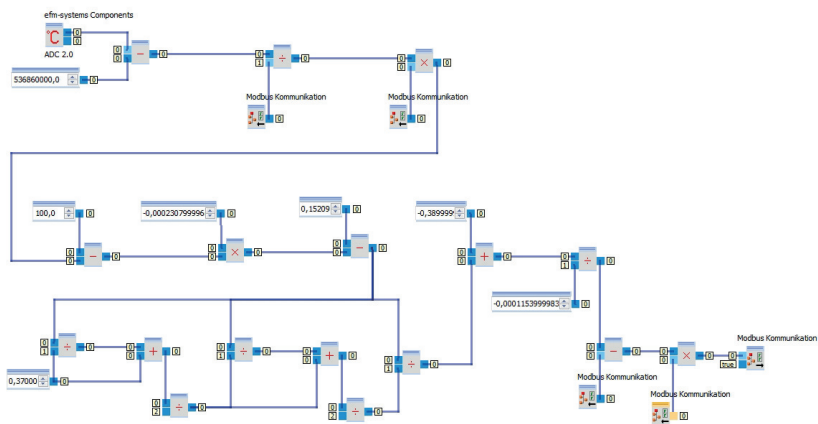


Abbildung 7.10: Vollständiges Programm zur Messwertberechnung der Temperatur in EasyLab

Zusammenfassend ist festzustellen, dass der Aufwand für die Erstellung und den Test der Firmware unter Berücksichtigung bereits vorhandener Programmteile (Bootloader und Modbus Kommunikation) ca. 2 AT beträgt. Die geforderten Eigenschaften des Systems können in der aktuellen Ausbaustufe von EasyLab nur zu 90% erfüllt werden. Insbesondere die Mittelwertbildung (fehlerhafte Funk-

tion) sowie die Justierung und die Datenübertragung zusätzlicher Register über Modbus sind aus genannten Gründen noch nicht zufrieden stellend.

7.1.7 Vergleich Software

Ein Vergleich der beiden Vorgehen zur Erstellung der Firmware zeigt deutlich die Vor- und Nachteile beider Varianten. Während der konventionelle Ansatz umfangreiche Kenntnisse in der Programmierung und den Programmiersprachen voraussetzt, ermöglicht es EasyLab auch Nichtexperten, einfach und zielorientiert funktionalen Programmcode zu erstellen. Insofern die dafür erforderlichen Funktionen vorhanden sind oder sich durch die vorhandenen Funktionen abbilden lassen, ist es möglich, vollständige Codes zu erstellen und dies in einer deutlich kürzeren Zeit als beim konventionellen Vorgehen (2 Tage <-> 16 Tage).

Bei der Flexibilität sind dem System EasyLab jedoch Grenzen gesetzt, wie der geforderte und nicht realisierte Zugriff auf den EEPROM des ADC-Bausteins gezeigt hat. Des Weiteren waren für die Erstellung des Programms nur vergleichsweise einfache Funktionen erforderlich, das gesamte Programm ist im zeitlichen Ablauf unkritisch. Bei anspruchsvolleren Abläufen und Funktionen kommt EasyLab hier schnell an Grenzen. Nicht vorhandene Funktionalitäten können jedoch von Experten programmiert und hinzugefügt und das Programm somit sukzessive ausgebaut werden. Die Debugging Funktionalität von EasyLab erweist sich bei der Programmerstellung als sehr hilfreich und ist gleichzeitig einfach in der Bedienung. Nachteilig ist allerdings der dabei erforderliche zusätzliche Programmcode, der insbesondere bei Mikrocontrollern mit kleinerem Speicher sehr schnell zu einem zu großen Codeumfang führt.

Beim vorliegenden Beispiel des Temperaturmesssystems konnten die kompletten Modbusfunktionen sowie die Justierfunktion über Modbus aufgrund des zu großen Codeumfangs in EasyLab nicht mehr realisiert werden. Im direkten Ausführungsmodus ist eine Programmierung noch möglich, jedoch erschwert die nicht mehr zur Verfügung stehende Debugging Funktion erheblich und führt zu einer sehr hohen Anzahl von Programmierzyklen. Beim konventionellen Vorgehen war der Speicher nach Einbindung dieser Funktionen erst zu ca. 85% belegt.

Ein weiterer großer Vorteil von EasyLab ist der Rückgriff auf vorhandene und vor allem getestete Funktionsblöcke. Im Gegensatz hierzu erfordern die im konventionellen Ansatz erstellten Funktionen umfangreiche Tests, bzw. bereits während der Erstellung einen nicht unerheblichen Zeitaufwand für die Fehlersuche.

Kriterium	Konventioneller Ansatz	EasyLab
Zugang für Nichtexperten	1	3
Einfachheit der Programmerstellung	2	3
Flexibilität bei der Erstellung von Funktionen	3	2
Flexibilität bei Änderungen im Programm	2	3
Debugging	2	3
Widerverwendbarkeit von Programmteilen	2	3
Zeitaufwand	2	3
Erstellung neuer Funktionen	3	2
Speicherbedarf	3	1
	20	23

Tabelle 7.4: Vergleich der Softwarewerkzeuge

Zusammenfassend ist festzustellen, dass EasyLab sehr gute Voraussetzungen als universelles Programmierwerkzeug bietet, ein Ausbau der Funktionsbibliotheken und eine noch effizientere Codegenerierung für das Debugging jedoch unerlässlich sind.

7.1.8 Systemintegration in Stufe 1 der Entwicklungsmethodik

Das so entwickelte und aufgebaute System kann bereits in eine applikations-spezifische Bauform integriert und als vollständiges Messsystem aufgebaut werden. Im konkreten Fall des Temperaturmesssystems ist die Integration in ein Edelstahlrundgehäuse mit einem Steckeranschluss M12 gefordert.

Auf der Seite der Spannungsversorgung/Kommunikation erfolgt der direkte Übergang vom Stecker JST 5-polig auf den Sensorstecker M12. Hierfür steht ebenfalls ein bereits realisierter Stecker mit einem entsprechenden Connector zur Verfügung. Der Bausteinaufbau kann in ein Edelstahlrohr 20x1 eingeführt werden (Diagonale des Matrix-Steckers = 17,9mm), das eine Länge von 50mm hat. Ein Pt100 Einbaufühler wiederum, der auf das Edelstahlgehäuse aufgepresst werden kann, wird mit dem JST-SR04 Connector komplettiert und kann so auf den Sensoranschluss des Bausteinstapels angeschlossen werden.

Nach dem Verpressen der nun vorliegenden Teilkomponenten entsteht ein in sich geschlossenes System, das bereits allen wesentlichen Anforderungen entspricht und in dieser Bauform direkt in Endaufbauten integriert werden kann.

Für die Dokumentation des Systemaufbaus stehen auf elektronischer Seite die Datenblätter der einzelnen verwendeten Bausteine sowie für die Software das erstellte EasyLab Programm zur Verfügung.

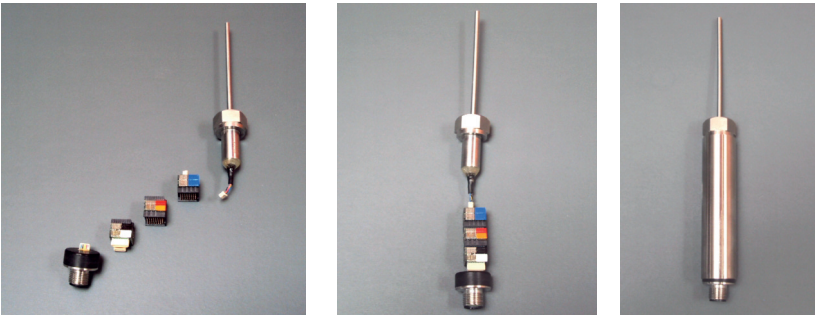


Abbildung 7.11: Links – Bausteine und Komponenten für Systemaufbau, Mitte - Montiertes Gesamtsystem, rechts – Komplettes Temperaturmesssystem mit Gehäuse

7.1.9 Aufbau des Systems nach Stufe 2 der Entwicklungsmethodik

Nachdem das Temperaturmesssystem in Stufe 1 erfolgreich aufgebaut, die Software erstellt und als Gesamtsystem getestet wurde, ist eine Überführung in eine Kleinserie und damit der Übergang von Stufe 1 zu Stufe 2 der Entwicklungsmethodik erforderlich.

Der Systemaufbau in Stufe 2 erfolgt durch Verwendung der gleichen Funktionsträger und direktes Verbinden mittels Lötén über Rahmenelemente. Für den Aufbau des Temperaturmesssystems sind folgende Funktionsträger erforderlich:

- Interfaceplatine BOT2.0
- Spannungsregler SRM2.0
- CPU-Platine CPU2.2
- A/D-Wandlerbaustein ADC2.0
- Interfaceplatine TOP2.0

Die verwendeten Funktionsträger sind dabei identisch denen, die in den Bausteinen BSRM2.0 (Funktionsträger BOT2.0 und SRM2.0), CPU2.2 und ADC2.0 (Funktionsträger ADC2.0 und TOP2.0) verwendet wurden. Zwischen den Funktionsträgern werden entsprechende Rahmenelemente angeordnet.

Die einzelnen Elemente werden durch Löten der außen liegenden Kontaktierungen direkt miteinander verbunden (siehe Kapitel 6.3.5). Es entsteht dabei ein System mit den Maßen 12,5mm x 12,5mm x 29mm. Anschließend kann die erstellte Software (wahlweise konventionell oder EasyLab) aufgespielt und das System so getestet werden. Zusätzlicher Programmier- oder Adaptionsaufwand in Hard- und Software besteht nicht. Da auch die Stecker identisch des Aufbaus in Stufe 1 sind kann die komplette Peripherie, d.h. der Einbaufühler mit den JST-Connector, 4-polig und der M12-Stecker mit dem JST-Connector, 5-polig weiterverwendet werden. Das so aufgebaute System kann wiederum in ein Edelstahlrundgehäuse \varnothing 20mm eingebaut werden, die Länge des Gehäuses kann bei Bedarf auf 40mm reduziert werden. Ein direkter Einsatz im industriellen Umfeld ist möglich.

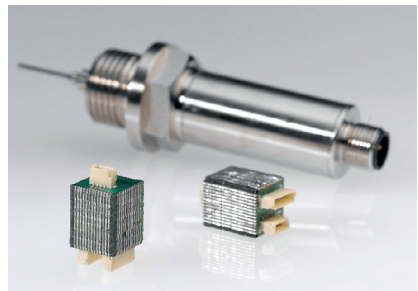
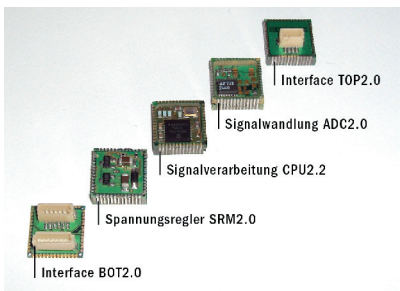


Abbildung 7.12: Links – Funktionsträger, rechts – Temperaturmesssystem nach Stufe 2 der Entwicklungsmethodik

Bei vorliegenden Funktionsträgern kann der Aufbau des Systems und somit der Übergang von Stufe 1 (Funktionsmuster) zu Stufe 2 (Prototyp/Kleinserie) innerhalb weniger Minuten erfolgen. Bei erforderlicher Nachbestellung einzelner Komponenten ist ein Aufbau des Systems bei entsprechenden Lieferzeiten innerhalb einer Woche möglich. Bezüglich der Dokumentation kann wie in Stufe 1 auf die vorliegenden Datenblätter sowie das EasyLab Programm für die Software zurückgegriffen werden.

7.1.10 Aufbau des Systems nach Stufe 3 der Entwicklungsmethodik

Bei steigenden Stückzahlen (>100) kommt der Mehraufwand des modularen Systemaufbaus bei den Kosten zum tragen. Daher bietet sich der Aufbau eines wirtschaftlich optimierten Systems auf einer Leiterplatte an.

Um eine Neuentwicklung oder umfangreichere Anpassentwicklung zu vermeiden, erfolgt ein Rückgriff auf die Schaltungsbibliotheken der einzelnen Funktionsträger und eine direkte Verknüpfung dieser zu einem durchgängigen Schaltplan für das Temperaturmesssystem. Für alle Funktionsträger stehen neben dem Datenblatt diese Schaltungsbibliotheken zur Verfügung. Zu beachten ist allerdings, dass diese nicht zwingend öffentlich zugänglich sind, d.h. dies ist Knowhow des Herstellers und kann mit zusätzlichen Kosten verbunden sein.

Zu beachten ist, dass bei einer Verknüpfung der Einzelfunktionen die Möglichkeit der Optimierung der Schaltung besteht. So sind z.B. im ADC-Baustein Funktionen zur Bausteinselektion enthalten, die bei einem direkten Aufbau auf einer Platine nicht mehr erforderlich sind. Jedoch führt die schaltungstechnische Optimierung in den meisten Fällen auch zu einer erforderlichen Anpassung der Software. Hier empfiehlt sich eine Kostenrechnung um zu ermitteln, ob bei den geplanten Stückzahlen der Mehraufwand für Schaltungs- und Softwareoptimierung durch die Einsparungen bei den Bauelementekosten wieder kompensiert werden kann. Im konkreten Fall des Temperaturmesssystems wird auf eine Anpassung der Schaltung verzichtet. Diese bietet sich aus schaltungstechnischer Sicht zwar an, da einige Funktionen, die aus dem modularen Aufbau resultieren, nicht mehr erforderlich sind, aber diese Anpassung würde zwangsläufig einen Eingriff in die getestete und freigegebene Software bedeuten. Da in diesem konkreten Fall das Einsparpotenzial der Bauelemente max. 50 Eurocent beträgt und gleichzeitig die Softwareanpassungen und Tests mit bis zu 5 Arbeitstagen kalkuliert werden (die Änderungen werden nicht mehr durch die vorhandenen Funktionsblöcke in EasyLab unterstützt, eine Erstellung ergänzender Funktionen ist zwingend erforderlich), wäre diese Änderungen erst bei Stückzahlen >5.000 wirtschaftlich.

Zurückkehrend zum konkreten Aufbau der Leiterplatte ist festzustellen, dass die einzelnen Funktionsblöcke so in zwei Ebenen angeordnet werden können, dass eine Leiterplatte mit den Abmaßen von 17,5mm x 40mm entsteht. Für den Anschluss des Pt100 wird wiederum der 4-polige JST Connector vorgesehen, für den Anschluss von Spannung und Versorgung werden 4 Durchkontaktierungen

geplant, um hier eine größere Flexibilität bei anzuschließenden Steckervarianten zu haben. Für die ergänzenden Komponenten bedeutet dies, dass wiederum der gleiche Einbaufühler verwendet werden kann, ein M12 Stecker mit offenen Litzenenden sowie das gleiche Rundgehäuse mit Außendurchmesser 20mm wie bisher und einer Länge von 50mm. Die kürzere Länge von 40mm des Aufbaus in Stufe 2 kann in dieser Bauform nicht erreicht werden.

Aufgrund der unveränderten Schaltung kann die bereits erstellte Software ohne weitere Änderungen aufgespielt, getestet und verwendet werden. Die so gestaltete Platine ist für eine Serienfertigung geeignet, auf allen üblichen Fertigungslinien bestückbar und mit herkömmlichen Testsystemen gut prüfbar.

Das im Gehäuse integrierte Temperaturmesssystem verfügt so in allen 3 Entwicklungsstufen über die gleiche Form und unterscheidet sich äußerlich nur in der Länge des Gehäuses.

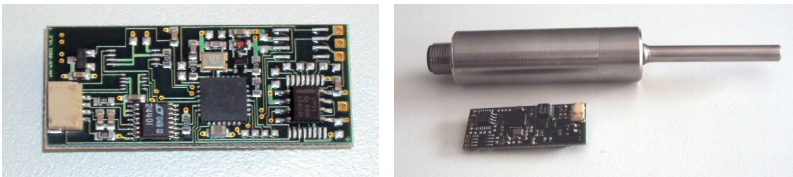


Abbildung 7.13: Elektronik des Temperaturmesssystems nach Stufe 3 der Entwicklungsmethodik

Der Zeitaufwand für den Übergang von Stufe 2 zu Stufe 3 setzt sich somit aus der Zusammenführung des Schaltplans (1-2 AT), der Erstellung des Layouts (5 AT), der Bestückung, Inbetriebnahme und des Test des Systems (5 AT) zusammen. Inklusiv der erforderlichen Durchlauf- und Wartezeiten (z.B. für die Leiterplatten und Beschaffung der Bauelemente 10 AT) ist von einer Gesamtlaufzeit der Überführung von Stufe 2 zu Stufe 3 von ca. 22 AT auszugehen.

7.2 Pneumatikzylinder

7.2.1 Allgemeine Anforderungen

Das zweite Beispiel ist der Aufbau eines Funktionsmusters eines Sensor-/Aktorsystems mit integrierter Regelung. Als Beispielsystem wird ein positionsge-
regelter Pneumatikzylinder gewählt, der die folgenden Anforderungen erfüllen soll:

- Verstellbereich: 50mm
- Positioniergenauigkeit: 2% FS (Fullscale) = $\pm 1,0\text{mm}$
- Ansteuerung der Kammern mittels 3/2-Wege-Ventilen
- Druck kann variabel zwischen 1bar und 5bar voreingestellt werden
- Vorgabe des Zielwertes im Programm oder über Modbus

Aus diesen Bedingungen ergibt sich das in Abbildung 7.14 dargestellte stark vereinfachte Modell des technischen Aufbaus.

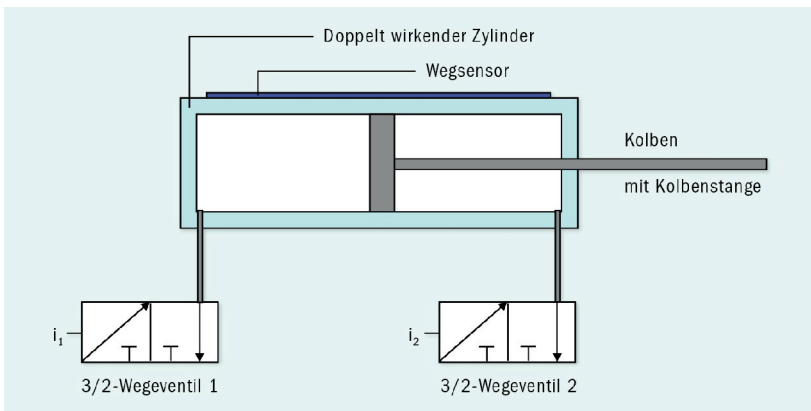


Abbildung 7.14: Technisches Modell

Für den Pneumatikzylinder und die Ventile soll auf marktverfügbare Komponenten zurückgegriffen werden. Die Elektronik des Systems soll soweit möglich auf Grundlage vorliegender Bausteine aufgebaut werden. Bezüglich der zu erstellenden Software wird ausschließlich der Ansatz mittels EasyLab gewählt.

7.2.2 Modell des Regelkreises und Regelstrategie

Vor der Auswahl der Komponenten wird ein Modell des Regelkreises (siehe Abbildung 7.15) erstellt, um die Anforderungen an Pneumatik, Elektronik und Software weiter detaillieren und die Störgrößen erfassen zu können.

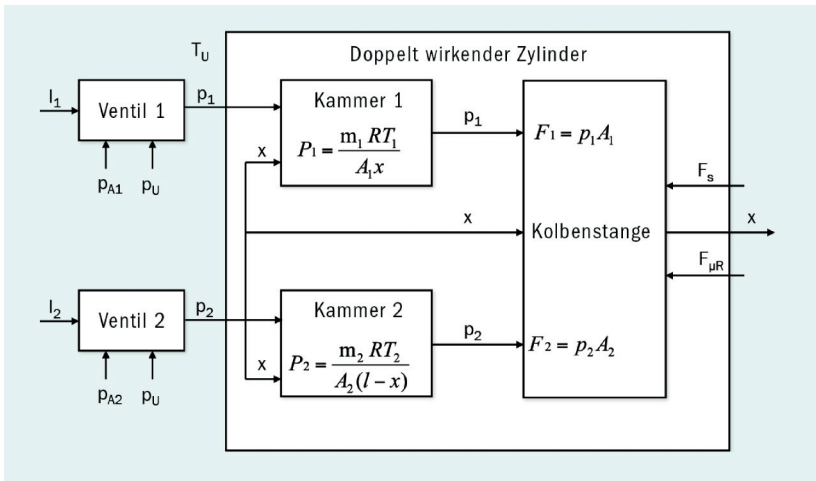


Abbildung 7.15: Blockschaltbild der Strecke mit Stelleinrichtung

An den 3/2 Wegeventilen liegt der Arbeitsdruck p_{A1} und p_{A2} sowie der Umgebungsdruck p_u an. Über die Schaltfunktion (Strom I_1 und I_2) des Ventils kann der Druck in den Kammern des Pneumatikzylinders (p_1 und p_2) somit zwei Zuständen annehmen, p_a und p_u . Für die Zylinderkammern gilt die Zustandsgleichung

$$pV = mRT \quad 7-5$$

mit

p – Druck in der Kammer

V – Volumen der Kammer

m – Masse des Gases

R – Gaskonstante

T – Temperatur des Gases

Es gilt:

$$V_1 = A_1 x \quad 7-6$$

$$V_2 = A_2 (l - x) \quad 7-7$$

A – Fläche am Kolben, auf die der anliegende Druck einwirkt

x – Stellweg der Kolbenstange

l – Länge der Kammer

Durch die einseitig angeordnete Kolbenstange des Zylinders ist die mit Druck beaufschlagte Fläche in den beiden Kammern verschieden, es gilt somit

$$A_1 \neq A_2 \quad 7-8$$

Daraus folgt, dass bei gleichem anliegenden Druck und konstanter Temperatur in beiden Kammern die auf den Kolben wirkenden Kräfte F_1 und F_2 unterschiedlich sind. Das elektrisch betätigte Ventil wiederum kann in zwei Zuständen betrieben werden, geöffnet und geschlossen. Über die Einschaltdauer kann das System somit geregelt werden, wobei zu beachten ist, dass die Schaltzeiten (Ein/Aus) sowie die maximal mögliche Schaltfrequenz als Parameter die Dynamik des geschlossenen Regelkreises begrenzen.

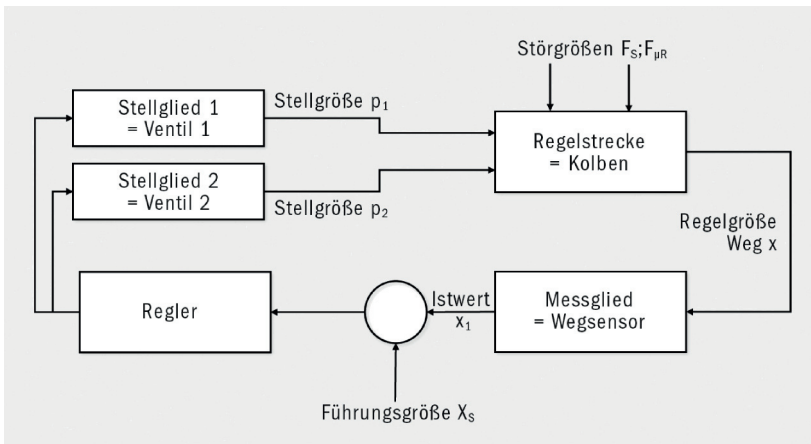


Abbildung 7.16: Blockschaftbild des Regelkreises

Darüber hinaus sind als offensichtliche Störgrößen eine auf die Kolbenstange wirkende Prozesskraft F_S sowie die Reibkraft $F_{\mu R}$ der Kolbenstange, die dem Stick-Slip Effekt unterliegt, zu beachten.

Aus diesen Größen kann somit das Blockschaltbild (Abbildung 7.16) des Regelkreises abgeleitet werden.

Die Hard- und Softwareumsetzung des Reglers soll dabei unter Nutzung von EasyKit und EasyLab erfolgen.

7.2.3 Funktionsstruktur und Teilfunktionen

In einem ersten Schritt erfolgt, unabhängig von der einzusetzenden Entwicklungsmethodik, die Zerlegung des geplanten Gesamtsystems in seine Teilfunktionen. Dabei ist die Zielstellung zu berücksichtigen, die komplette Sensor-/ Aktorkette in einem System abzubilden. Hierfür ist von den in Tabelle 7.5 und Abbildung 7.17 dargestellten grundlegenden Funktionen auszugehen.

Funktion	Beschreibung
Spannungsversorgung/ -adaption	Anpassung der anliegenden externen Spannung (12VDC oder 24VDC) an die systeminterne Spannung sowie an die erforderlichen Aktorspannungen
Kommunikation	Bereitstellung der hardwaretechnischen Voraussetzungen für eine RS485 Kommunikation
Anschluss Spannung und Kommunikation	Verbindungsmöglichkeit zu externen Komponenten über Sensorstecker M12
Signalverarbeitung	Bereitstellung der softwaretechnischen Voraussetzungen für die Kommunikation Berechnung der Position des Pneumatikzylinders Ermittlung der Abweichung von Soll-/Istposition und daraus abgeleitet Ansteuerung der Ventile zur Korrektur der Position
Signalwandlung Sensor	Wandlung des analogen Sensorsignals in ein digitales Signal zur weiteren Verarbeitung
Ansteuerung Ventil	Wandlung der digitalen Vorgabe der Signalverarbeitung in ein Ansteuersignal für die Ventile
Anschluss Sensor	Verbindungsmöglichkeit zu externem Wegsensor
Ansteuerung Ventil	Verbindungsmöglichkeit zu externen Ventilen

Tabelle 7.5: Aufstellung der Teilfunktionen des Pneumatikzylinders

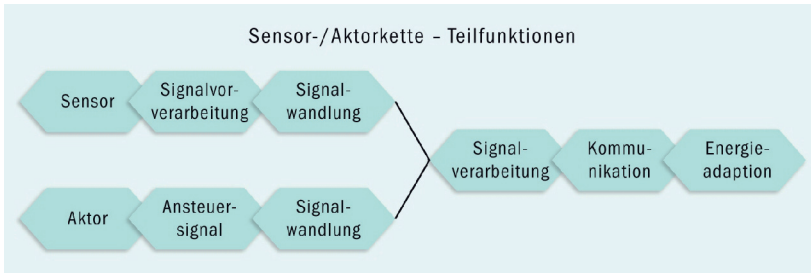


Abbildung 7.17: Sensor-/Aktorkette

7.2.4 Aufbau des Systems nach Stufe 1 der Entwicklungsmethodik

In einem weiteren Schritt erfolgen die detaillierte Beschreibung der Einzelfunktionen und die Überprüfung auf vorhandene Funktionsblöcke in der Entwicklungsmethodik EasyKit.

Pneumatikzylinder, Ventile und Wegmessung

Als Basis fungieren ein Pneumatikzylinder, an den 2 Ventile angeschlossen werden, sowie ein für den Zylinder passender Wegsensor. Diese werden wie folgt für die Aufgabe ausgewählt.

Als Pneumatikzylinder kommt ein doppelwirkender Kompaktzylinder mit ausfahrender Kolbenstange vom Typ ADN-12-50 der Firma Festo [Festo 2012_ADN] zum Einsatz. Der Zylinder ist mit einem passenden Profilstück zu verlängern, in dem wiederum die Elektronik zu integrieren ist, um im Ergebnis einen sehr kompakten Zylinder mit integrierter Ansteuereinheit zu erhalten.

Als Ventile kommen 3/2-Wege Ventile der Firma Festo vom Typ MHA1 [Festo 2012_MHA] zum Einsatz. Die ausgewählten Ventile sind im nicht bestromten Zustand geschlossen, haben einen zulässigen Druck von 0...6 bar und werden mit einer Betriebsspannung von 24V geschaltet. Für die Regelung zu beachten ist außerdem noch die maximale Schaltzeit von 4ms (gilt sowohl für das Ein- als auch das Ausschalten), die nicht unterschritten werden darf, sowie die maximale Schaltfrequenz, die mit 20Hz laut Datenblatt angegeben ist.

Als Wegsensor wird der SMAT-8E-S50 [Festo 2012_SMAT] eingesetzt. Dieser kann direkt in die T-Nut des Zylinders eingesetzt werden. Der Sensor verfügt bereits über eine integrierte Signalvorverarbeitung, so dass am Ausgang des Sensors

ein 0...10V Signal anliegt. Zu beachten ist hierbei jedoch, dass die Signalpegel 0...2V mit Statusinformationen belegt sind (0 = kein gültiges Signal, 1V und 2V = außerhalb des gültigen Messbereichs), so dass nur für den Bereich > 2V bis 10V ein gültiges Messsignal ausgegeben wird. Dabei wird die Empfindlichkeit des analogen Ausgangs laut Datenblatt mit 0,152V/mm und die Reproduzierbarkeit mit 0,064mm angegeben. Der Positionstransmitter verfügt wahlweise über einen 4-poligen Sensorstecker M8 oder über ein Kabel mit offenen Enden.

Signalerfassung

Es ist gefordert, die Position des Pneumatikzylinders über die gesamte Länge (50mm) mit einer Genauigkeit von $\pm 1,0\text{mm}$ zu regeln. In EasyKit stehen Bausteine mit A/D-Wandlerfunktion mit 10bit, 12bit und 24bit Auflösung zur Verfügung. Für die geforderte Genauigkeit von 1,0mm sollte ein Signal mit einer Auflösung von mindestens 0,1mm (=500 digits) zur Verfügung gestellt werden. Hierfür sind alle vorliegenden A/D-Wandler geeignet, vorausgesetzt der Messbereich von 0...10V wird vollständig auf den Eingangsbereich des A/D-Wandlers skaliert.

Da alle A/D-Wandlereingänge für einen Messbereich von 0...5V DC oder kleiner ausgelegt sind, ist es zwingend erforderlich einen Adaptionsteil einzusetzen, der die Ausgangsspannung des Wegsensors von 0...10V adaptiert und idealerweise gleichzeitig eine Schutzfunktion für Verpolung und Überspannung enthält. Hierfür stehen die Bausteine der Kategorie ADU2.X zur Verfügung. Mit diesen Bausteinen kann je nach Variante das Verhältnis zwischen Eingang und Ausgang eingestellt werden. Für den vorausgewählten A/D-Wandlereingang ist somit ein Verhältnis von 2:1 erforderlich. Dieses Verhältnis bietet der Baustein ADU2.4, bei dem gleichzeitig zwei Eingangskanäle zur Verfügung stehen.

Ventilansteuerung

Für die Ansteuerung der zwei Ventile sind zwei getrennte Ausgänge an der Elektronik erforderlich. Diese müssen in der Lage sein, die Betriebsspannung von 24V DC zu schalten (Ein/Aus).

In EasyKit stehen für diese Funktion prinzipiell zwei Bausteintypen zur Verfügung, die Kategorie DIO und die Kategorie ILT. Beim DIO handelt es sich um digitale Eingang-/Ausgangsbusteine. Mit dem DIO24 liegt ein Baustein vor, der die 24V des Ventils schalten kann. Bei der Kategorie ILT handelt es sich um induktive Lasttreiber, die speziell für den Betrieb von Relais oder Magneten ausgelegt sind. Dieser Bausteintyp kann Spulenströme von 65mA bis 300mA schalten und hat nach 100ms eine automatische Lastabsenkung zur Reduzie-

rung der Leistungsaufnahme im Haltebetrieb integriert. Somit ist dieser Typ ideal für die Ansteuerung des Ventils MHA1 geeignet.

Um zwei Ventile ansteuern zu können, sind jedoch zwei Ausgänge erforderlich. Hierfür liegen mit dem ILT2.1 und ILT2.2 zwei Bausteine vor, die unterschiedliche Ausgangspins verwenden (siehe Kapitel 6.1) und über getrennte Anschlusspins vom CPU-Baustein angesteuert werden können.

Signalverarbeitung und Kommunikation

Für die Signalverarbeitung und Kommunikation stehen in EasyKit die bereits beschriebenen CPU-Bausteine zur Verfügung, die beide Funktionen direkt miteinander verbinden. Zur Auswahl eines geeigneten CPU-Bausteins ist es erforderlich, die geforderten Berechnungsroutinen zu prüfen und den Ressourcen der CPU gegenüberzustellen.

Im vorliegenden Beispiel sind folgende grundlegenden Operationen erforderlich:

- Wandlung des ADC-Rohwertes in den äquivalenten Wert der Eingangsspannung
- Umrechnung der Spannung in die Position des Wegsensors
- Mittelwertbildung (bei Bedarf)
- Vorgabe eines Sollwertes (über Modbus oder direkt im Programm)
- Ermittlung der Differenz zwischen Soll- und Istwert
- Einstellung der Position über einen geeigneten Regler
- Ansteuerung der Ventile
- Kommunikation mittels Modbus RTU

Auf Basis dieser Prüfung erfolgte wiederum die Auswahl des Bausteins CPU2.2 für die Signalverarbeitung und Kommunikation. Wie bereits beschrieben verfügt der Baustein über einen 8bit Mikrocontroller (PIC18F2520) mit externem Quarz sowie einem integrierten RS485 Treiber. Darüber hinaus sind mindestens zwei A/D-Wandlereingänge (10bit) nach außen geführt und stehen über den internen Bus zur weiteren Verwendung zur Verfügung.

Die Ressourcenprüfung ergab eine sehr hohe Auslastung der CPU2.2, so dass eigentlich der Übergang zum nächst größeren Typ erfolgen sollte. Dieser befand

sich jedoch zum Zeitpunkt der Erstellung dieser Arbeit noch in Entwicklung. Daher wurden die Arbeiten trotz entsprechender Bedenken mit der CPU2.2 fortgesetzt.

Spannungsadaption

Die Anforderungen an die Spannungsadaption sind die Transformation einer externen Eingangsspannung von 24VDC (Forderung aufgrund der ausgewählten Ventile) auf die interne Systemspannung sowie die Durchleitung von 24V DC zur Versorgung der Ventile. Die ausgewählten Bausteine CPU2.2, ADU2.4 und ILT2.X arbeiten mit 5VDC bzw. mit 24V DC.

Für diese Anforderungen steht der Baustein SRM2.3 zur Verfügung. Dieser ist in der Grundkonfiguration analog dem bereits ausgewählten Baustein SRM2.0 (siehe Kapitel 7.1.3), jedoch ist bei diesem Baustein die externe Versorgungsspannung zur weiteren Verwendung auf den internen Bus geführt. In Verbindung mit einer Interfaceplatine kann der Baustein so komplettiert werden, dass ein direkter Anschluss über Stecker an externe Komponenten ermöglicht wird.

Anschlussinterface

Die externen Komponenten sind über einen Stecker oder eine integrierte Klemmleiste direkt an das aufzubauende System anzuschließen. Da ein entsprechendes Interface in EasyKit nicht vorliegt, ist dieses anwendungsspezifisch auszulegen und aufzubauen.

Somit können in diesem Beispiel alle erforderlichen elektronischen Grundfunktionen des Gesamtsystems mit Funktionen aus EasyKit abgedeckt werden, ein direkter Systemaufbau ist möglich. Für die eigentliche Integration in den Pneumatikzylinder ist jedoch ein spezifisches Anschlussinterface auszulegen.

Zuweisung der EasyKit Funktionsblöcke zu den Teilfunktionen

In Abbildung 7.18 werden die Teilfunktionen des Systems, deren Überführung in Funktionsblöcke von EasyKit und damit verbunden in die konkreten, vorliegenden Funktionsbausteine von EasyKit dargestellt.

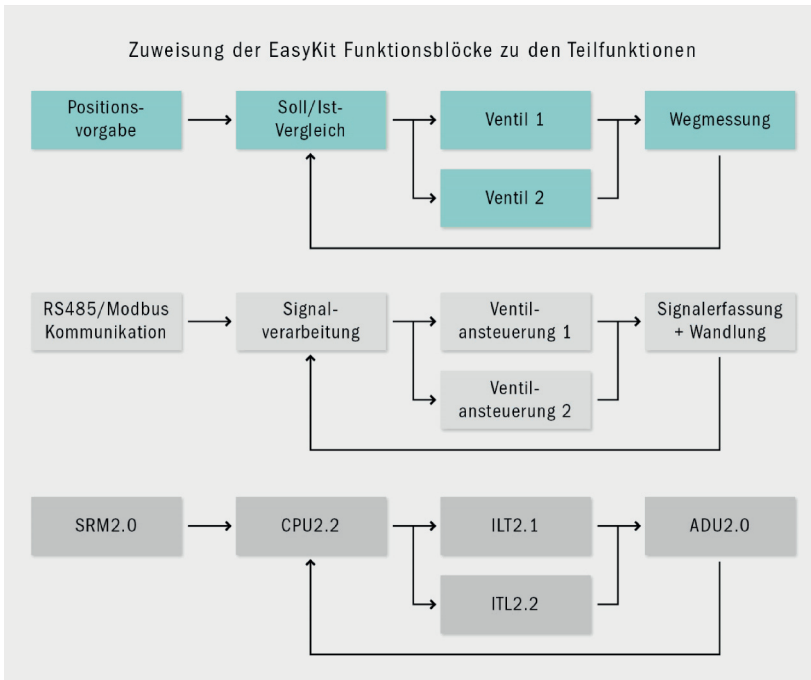


Abbildung 7.18: Blockschaltbild der Zuordnung der Bausteine zum geplanten Systemaufbau/Regelkreis

Demzufolge sind für den Aufbau des Regelkreises folgende Funktionen erforderlich:

- SRM2.3: Spannungsregelung der Versorgungsspannung 24V auf die internen Systemspannungen von 5V für die CPU und 5V für den A/D-Wandler und die Sensorversorgung
- CPU2.2: Modbus-Kommunikation, Ansteuerung der Ventile, Auswertung des Messsignals des Wegsensors, Soll-/Ist-Wert-Vergleich, Regler
- ILT2.x: Treiberausgang für zwei induktive Lasten
- ADU2.0: 2-kanalige Schutzbeschaltung für die 10bit A/D-Wandlereingänge der CPU

7.2.5 Auslegung des Anschlussinterface

Nachdem die elektronischen Teilfunktionen bereits alle als Funktionen in Easy-Kit zur Verfügung stehen, ist der nächste Schritt die Auslegung eines Anschluss-interfaces, welches eine direkte Integration der Bausteine in ein an den Zylinder angeschlossenes Profilstück ermöglicht. Die beiden hier vorgestellten Anschluss-interfaces wurden durch die Firma digiraster Tetzner GmbH im Rahmen des Projekts EasyKit konstruiert und auch gefertigt.

Die anzuschließenden Sensoren und Aktoren liegen dabei hinsichtlich ihres Anschlusses in unterschiedlichen Ausführungsformen vor. Das auszulegende Interface muss somit Anschlüsse für die in Tabelle 7.6 aufgeführten Komponenten bieten:

Komponente	Anschluss werksseitig	Anschluss alternativ
Spannungsversorgung und Kommunikation	Kabel mit Sensorstecker M8 – 4-polig	Kabel mit offenen Enden – Klemmleiste
Wegsensor	Sensorstecker M8 – 4-polig	Kabel mit offenen Enden – Klemmleiste
Ventil 1	Sensorstecker M8 – 3-polig	Kabel mit offenen Enden – Klemmleiste
Ventil 2	Sensorstecker M8 – 3-polig	Kabel mit offenen Enden – Klemmleiste

Tabelle 7.6: Komponenten und Anschlussarten

Um im Entwicklungsprozess eine hohe Flexibilität zu haben und gleichzeitig für die Integration einen sehr kompakten Aufbau zu erreichen, wurde für beide Anschlussversionen eine Lösung aufgebaut.

Für die Variante mit Steckeranschlüssen wurde eine Aufbauform gewählt, bei der auf der einen Seite der Baustein stapel über den Matrixstecker aufgesetzt werden kann und auf der anderen Seite direkt die erforderlichen M8 Stecker aufgelötet werden können. Aufgrund des extrem kleinen Bauraums war es erforderlich, den M8 Stecker für die Spannungsversorgung/Kommunikation mittels Starr-Flex-Technologie seitlich anzuordnen, während die Stecker für die Sensoren und Aktoren stirnseitig angeordnet sind. Der Aufbau wurde direkt in ein Aluminium Frästeil integriert, das wiederum als Abschlussdeckel auf das Aluminiumprofil aufgesetzt werden kann.

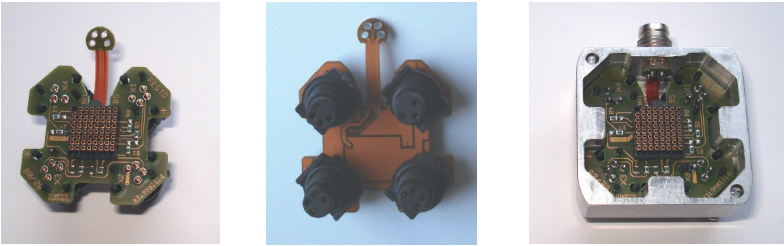


Abbildung 7.19: Lösung 1 des Anschlussinterface: Links und Mitte Starr-Flex-Leiterplatte, rechts Leiterplatte mit Baueinstapel integriert in Abschlussdeckel

Bei der zweiten Lösung war das Ziel eine höhere Flexibilität für den Anschluss von Sensoren und Aktoren bzw. weiterer möglicher Komponenten. Daher wurde der Aufbau aus zwei Komponenten realisiert. Auf einem Basisboard sind Klemmleisten angeordnet, bei dem die unterschiedlichen Ein- und Ausgänge des Baueinstapels offen zugänglich sind. An einer Stirnseite sind die externen Anschlussstecker angeordnet (USB und Versorgung/RS485 als 4-poliger M8 Rundsteckverbinder), auf der anderen Seite ist der Übergang auf den Matrix-Stecker für die Bausteine. Hierfür ist eine rechtwinklige, doppelreihige Stiftleiste angebracht, so dass eine Platine mit dem Matrixstecker (quasi als Grundposition) angebracht werden kann. Auch diese Variante wurde so aufgebaut, dass eine direkte Integration in das Anschlussprofil des Pneumatikzylinders möglich ist.

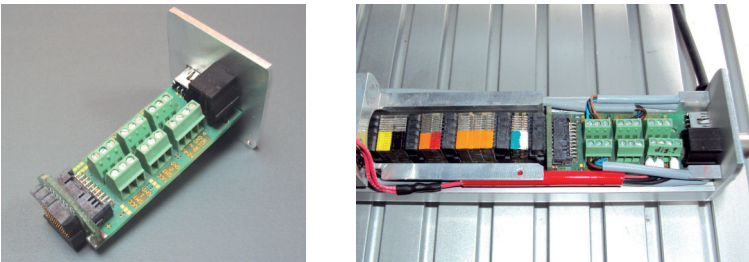


Abbildung 7.20: Lösung 2 des Anschlussinterface: links Grundaufbau, rechts Aufbau mit Baueinstapel

Die realisierte Platine mit den Klemmleisten kann darüber hinaus als Funktionskomponente für andere EasyKit Aufbauten eingesetzt werden, da die auf der Klemmleiste angebotenen Anschlüsse die wesentlichen Ein- und Ausgänge der Bausteine repräsentieren.

7.2.6 Aufbau des integrierten Pneumatikzylinders mit den EasyKit Bausteinen

Der integrierte Pneumatikzylinder besteht im Wesentlichen aus zwei Systemteilen, dem Zylinder mit Sensoren und Aktoren sowie dem Aluminiumprofil mit der integrierten Elektronik. Der Aufbau der Pneumatik mit Sensor und Aktor erfolgte hierbei direkt durch die Firma Festo Didactic. In der ersten Aufbauvariante wurden die beiden Ventile vom Typ MHA1 über einen Adapter direkt auf die pneumatischen Anschlüsse des Zylinders aufgesetzt. Der Wegsensor SMAT-8E-S50 wurde direkt in die Nut eingebracht und deckt mit seiner aktiven Länge den Wegbereich des Zylinders ab. Die Anschlussstecker des Wegsensors sowie der Ventile verfügen direkt über angeschlossene Sensorstecker M8.

Das Aluminiumprofil ist 30mm lang und wird über zwei Schrauben direkt an den Pneumatikzylinder angeschlossen. Das Kernloch des Profils hat einen Durchmesser von 20mm. In dieses Kernloch wiederum können die Bausteine des Anschlussinterfaces direkt eingeführt werden.

Aufgrund des durchgehenden Bussystems über die außen liegenden Kontakte stehen alle Anschlüsse an beiden Enden des Bausteinstapels zur Verfügung. Somit können alle Bausteine (SRM2.3, CPU2.2, ADU2.0, ILT2.1 und ILT2.2) direkt übereinander gesetzt werden und dieser Stapel wiederum direkt auf den Matrixstecker des Anschlussinterfaces. Der Bausteinstapel wird im Kernloch versenkt und das Anschlussinterface auf das Aluminiumprofil aufgeschraubt. Somit entsteht ein komplett geschlossener Aufbau.

Abschließend werden die einzelnen Sensoren und Aktoren sowie die Spannungsversorgung und die Kommunikation angeschlossen. Der Wegsensor wird einzeln auf einen 3-poligen Sensorstecker M8 geführt (Versorgung 5V DC, Ground und Messsignal). Die beiden Ventile wiederum werden ebenfalls auf zwei einzelne 3-polige Sensorstecker geführt (Versorgung 24V DC, Ground, Schaltsignal). An den seitlich angeschlossenen 4-poligen Sensorstecker M8 wird ein Kabel mit der Spannungsversorgung für das System (24V DC) und der RS485 Kommunikation angeschlossen. Im Ergebnis entsteht ein sehr kompakter, miniaturisierter Aufbau.



Abbildung 7.21: Pneumatikzylinder mit Anschlussinterface 1

In der zweiten Aufbauvariante ist der Grundaufbau des pneumatischen Teils mit den Ventilen nahezu identisch. Zur besseren Adaption wurde ein Kunststoffblock gefertigt, der zwischen Pneumatikzylinder und Ventilen sitzt und einen einfacheren Anschluss der pneumatischen Komponenten ermöglicht.

Das Aluminiumprofil ist in dieser Ausführung 100mm lang und halbseitig geöffnet. So sind sowohl die Bausteine als auch die Anschlussklemmen für die Sensoren und Aktoren gut zugänglich. An den Anschlussblock (siehe auch Abbildung 7.20) werden direkt die erforderlichen Bausteine (identisch dem ersten Anschlussinterface) angesteckt. Die Kabel aller Sensoren und Aktoren werden im Profil geführt und über die Klemmleiste angeschlossen. Dieser Aufbau bietet eine höhere Flexibilität als das erste Anschlussinterface, da bis zu drei analoge und zwei digitale Sensoren sowie zwei Aktoren angeschlossen werden können und so das Maximum der über den Bus der Bausteine zur Verfügung gestellten Signale ausgenutzt werden kann. Gleichzeitig bieten die Klemmleisten eine sehr einfache Austauschmöglichkeit. Über eine Kunststoffschale wird das Profil geschlossen.

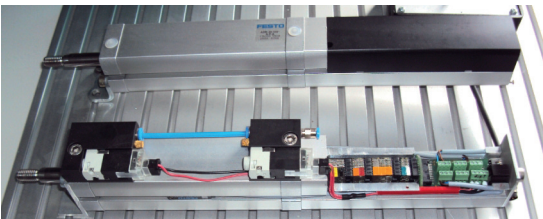


Abbildung 7.22: Pneumatikzylinder mit Anschlussinterface 2

7.2.7 Erstellung der Firmware mittels EasyLab

Die Umsetzung der Firmware mittels EasyLab basiert auf der Grundlage des erstellten Blockschaltbildes (Abbildung 7.16) des Regelkreises sowie der Zuweisung der EasyKit Bausteine in Verbindung mit den dafür zur Verfügung stehenden EasyLab Funktionen. In einem ersten Schritt erfolgt die Prüfung der durch die Hardware vorgegebenen Funktionen:

Funktion	Hardware	EasyKit Baustein	EasyLab Funktion
Wegsensor	Wegsensor	ADU2.0	Analoger Eingang RA0
Wegvorgabe	-	-	Eingabefenster Konstante
Schalten Kammer 1	Ventil 1	ILT2.1	ILT C1
Schalten Kammer 2	Ventil 2	ILT2.2	ILT C2

Tabelle 7.7: Hardwareabhängige Funktionsblöcke in EasyLab zur Erstellung der Firmware für den Pneumatikzylinder

In einem zweiten Schritt erfolgt die Erstellung des Teilprogramms für die Auswertung des Wegsensors. Bei diesem ist wie bereits beschrieben zu beachten, dass zwar ein 0...10V Signal geliefert wird, der eigentliche Messbereich jedoch nur von 2...10V geht. Das eigentliche Messsignal ist somit wie folgt zu ermitteln:

$$y = ax + b$$

7-9

mit

y – gemessener Weg des Sensors in mm

x – Messwert des Wegsensors in V

sowie den Konstanten

a = 6,25mm/V

b = -12,5mm

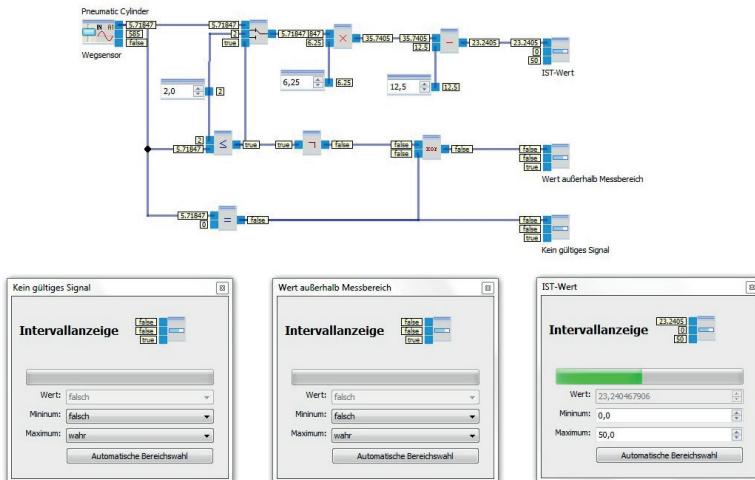


Abbildung 7.23: Teilfunktion für die Berechnung der Position des Pneumatikzylinders

Mit dem in Abbildung 7.23 dargestellten Teilprogramm können alle geforderten Bedingungen zur Auswertung des Wegsensors erfüllt werden. Bei einem Wert kleiner 2V werden die Anzeigen „Kein gültiges Signal“ oder „Wert außerhalb Messbereich“ gesetzt, erst ab 2V schaltet der Zweiwegeschalter das Signal durch und dieses fließt in die Verarbeitungsroutine nach Formel 6-9 ein.

Ein Test am Aufbau des Pneumatikzylinders bestätigt die Funktion. Gleichzeitig konnte mit diesem Test eine Signalstabilität von $\pm 0,2\text{mm}$ ermittelt werden. Für eine Signalberuhigung steht in EasyLab die Funktion „Glätter“ zur Verfügung, mit der ein Mittelwert über die letzten 3 Werte gebildet werden kann. Eine Mittelung über mehr Werte wäre wünschenswert, lässt sich aber über die aktuell vorliegende Funktion nicht einstellen. Trotz Einsatz der Mittelwertfunktion kann nur eine maximale Signalstabilität von $\pm 0,15\text{mm}$ erreicht werden (siehe Abbildung 7.24).

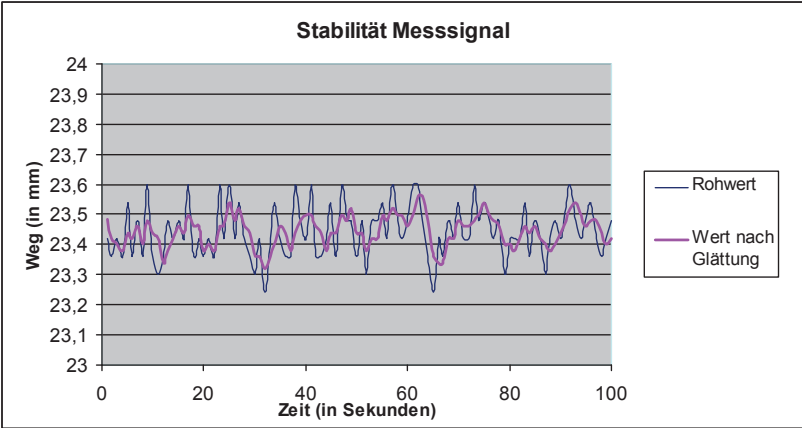


Abbildung 7.24: Signalstabilität des Messwertes in EasyLab mit und ohne Glättungsfunktion

Als nächstes wird die Regelabweichung durch einen Vergleich des berechneten Ist-Wertes mit dem vorgegebenen Soll-Wert ermittelt. Hierzu erfolgt einfach die Bildung der Differenz dieser beiden Werte in EasyLab.

Funktionsblöcke	Verfügbar	Benutzt
Funktionsblockbibliothek		
Regler		
Dreipunktregler	*	0
Zweipunktregler/Hystereseschleife	*	2
P-Regler	*	0
PI-Regler	*	0
PD-Regler	*	0
PID-Regler	*	0
Signalverarbeitung		
Signalgenerierung		
Konstanten		
Mathematik		
Verschiedenes		
Anzeige		
Variablen		
Pneumatic Cylinder		
efm-systems CPU (red/orange)		
efm-systems ADU (inline) (white/turquoise)		
Analoger Eingang RA0	0	1
Analoger Ausgang DA1	0	1

Abbildung 7.25: Funktionsblöcke in EasyLab zur Verwendung für den Pneumatikzylinder

In einem weiteren Schritt ist ein geeigneter Regler für diese Ausgabe auszuwählen. In EasyLab stehen hierfür als bereits fertig programmierte Funktionsblöcke stetige lineare Regler (z.B. PI oder PID-Regler) sowie unstetige Regler (z.B. Zweipunkt- oder Dreipunktregler zur Verfügung).

Bei den stetigen Reglern kann die Ausgangsgröße des Reglers jeden beliebigen Wert (0...100%) annehmen. Im Gegensatz hierzu ist bei unstetigen Reglern die Ausgangsgröße gestuft, bei einem Zweipunktregler sogar in nur zwei diskrete Werte.

Da als Stellglied Ventile zur Verfügung stehen, die nur durch zwei diskrete Werte angesteuert werden können (An/Aus), bietet sich somit die Verwendung eines Zweipunktreglers aus EasyLab an.

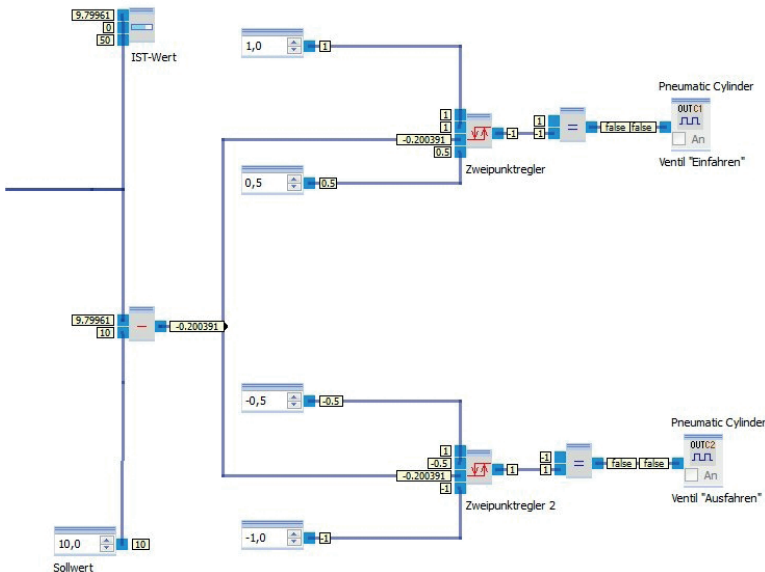


Abbildung 7.26: Zweipunktregler mit Toleranzgrenzen und Ventilen als Stellglieder in EasyLab

Im Ergebnis liegt ein Programm vor (Abbildung 7.26), bei dem je nach ermittelter Position beide Ventile angesteuert und bei Bedarf wechselseitig betrieben werden, um so den Kolben des Zylinders in die entsprechende Position zu führen. Der Test dieses Aufbaus ergab eine prinzipielle Funktionsfähigkeit, jedoch konnte

mit dieser Konfiguration die geforderte Genauigkeit von $\pm 0,5\text{mm}$ nicht erreicht werden. Bei einem Druck von 1,5bar lag die Einstellgenauigkeit bei $\pm 1,5\text{mm}$. Im Diagramm (Abbildung 7.27) ist eine Messung zum Anfahren einer Position von 25mm dargestellt, bei der die Zielposition mit einer Genauigkeit von $+1,3\text{mm}$ erreicht wurde.

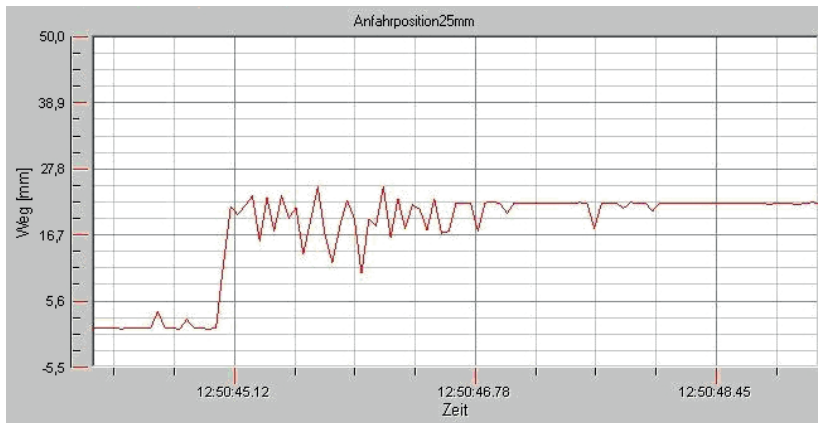


Abbildung 7.27: Plot des Anfahrens einer Position mittels Zweipunktreger

Um die Ursachen für die zu große Abweichung mit dem verwendeten Regler zu ermitteln, wurden in EasyLab verschiedene Programmversionen erstellt, um so einerseits den minimal möglichen Verstellweg zu ermitteln und andererseits das Anfahren der Position zusätzlich zu dämpfen.

Der erste Versuch war die Ermittlung des kleinsten Verstellweges. Hierfür wurde in EasyLab ein kleines Testprogramm erstellt, bei dem bei einer minimalen Öffnungszeit von 10ms der zurückgelegte Weg analysiert wurde. Kleinere Zeiten waren in EasyLab nicht einstellbar (Beschränkung des aktuell vorliegenden Programms). Jedoch ist bei der minimal einstellbaren Ansteuerzeit des Ventils der zurückgelegte Weg in der Regel größer 2mm, abhängig von der aktuellen Position des Zylinders. Zu berücksichtigen ist darüber hinaus, dass immer erst die Haftreibung des Kolbens überwunden werden muss, was als Störgröße auf den geschlossenen Regelkreis einwirkt.

In einem zweiten Versuch wurde ein Programm (Abbildung 7.28) erstellt, bei dem die Zielposition gedämpft angefahren wurde. Hierfür wurde in einem ersten

Unterprogramm der Zylinder stets in die Nulllage gefahren, von dieser Position aus in einem zweiten Unterprogramm das erste Ventil so lange geöffnet, bis ein definierter Abstand zur Zielposition unterschritten wird. Bei Erreichen dieser Bedingung erfolgt der Sprung in ein weiteres Unterprogramm, in dem das zweite Ventil hinzugeschaltet wird. Durch die größere Kraft aufgrund der höheren Anpressfläche in Kammer 1 wird der Kolben weiterbewegt, wobei durch das Umschalten des zweiten Ventils eine Dämpfung zu registrieren ist. Bei Erreichung der Zielgröße (Toleranzfeld) werden beide Ventile ausgeschaltet (Sprung ins vierte Unterprogramm).

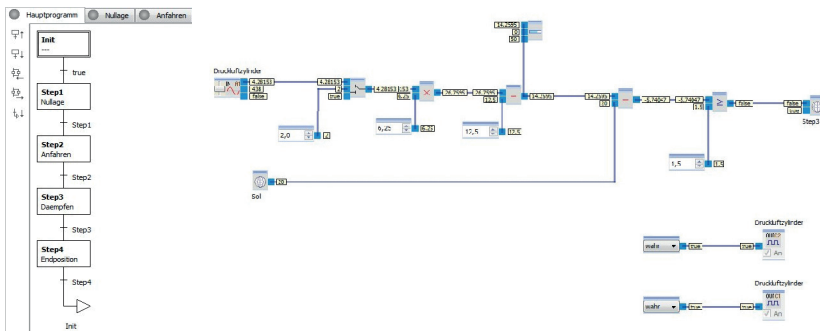


Abbildung 7.28: Ablaufprogramm und Unterprogramm 3 in EasyLab

Die praktischen Tests ergaben hier wieder eine große Abhängigkeit vom Weg. Im Bereich 0...30mm konnte eine Genauigkeit von 0,5...1mm erreicht werden, bei Einstellwerte über 30mm nahm der Fehler jedoch sprunghaft zu.

In Abbildung 7.29 ist zu sehen, wie im ersten Schritt die Nullposition angefahren, anschließend Ventil 1 geöffnet wird (starker Anstieg) und nach Öffnen des Ventils 2 sich der Anstieg der Kurve verringert (Dämpfung). Die Zielposition kann somit direkt angefahren werden.

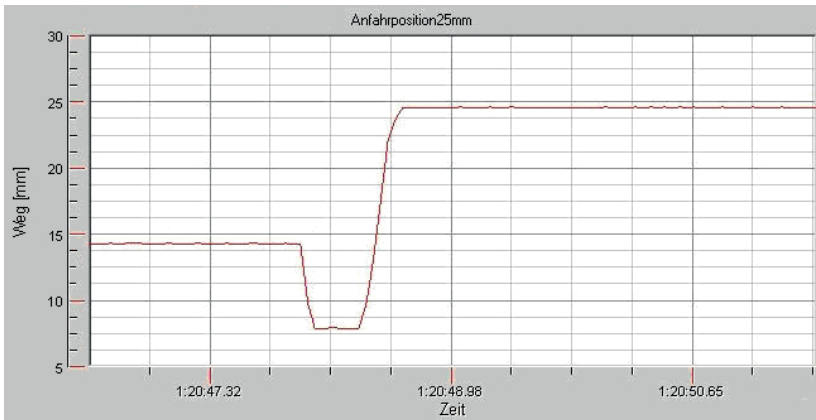


Abbildung 7.29: Plot des Anfahrens der Zielposition mit Dämpfung

Mit dieser Programmversion konnte eine Einstellgenauigkeit von $\pm 1,0\text{mm}$ im eingeschränkten Bereich von $0\ldots 30\text{mm}$ durch direktes Anfahren der Position erreicht werden. Bei Zielwerten größer 30mm fuhr der Zylinder stets über die geforderte Position. Hier wäre in Abhängigkeit der eingegebenen Zielposition eine Tabelle oder Funktion zu hinterlegen, die die Strecke der Dämpfung festlegt. Jedoch musste in dieser Programmversion festgestellt werden, dass die Ressourcen des verwendeten Mikrocontrollers für einen weiteren Ausbau des Programms nicht ausreichen. Ein Übergang auf einen leistungsfähigeren Baustein mit größerem Speicher ist somit für diese Aufgabenstellung zwingend erforderlich. Dieser Baustein befindet sich zum Zeitpunkt der Erstellung dieser Arbeit in Entwicklung und steht somit für weitere Untersuchungen derzeit noch nicht zur Verfügung.

7.2.8 Zusammenfassung Pneumatikzylinder

Der Aufbau des Pneumatikzylinders zeigt sehr gut die Möglichkeiten, aber auch die Grenzen des aktuell vorliegenden Systems auf. Dies gilt sowohl für die verwendete Hardware als auch das Programmierwerkzeug EasyLab.

Positiv hervorzuheben sind der schnelle Aufbau eines ersten Funktionsmusters und das Vorliegen aller Hard- und Softwarefunktionen in EasyKit und EasyLab. So war es in sehr kurzer Zeit möglich ein System aufzubauen, für das bei kon-

ventionellem Entwicklungsvorgehen sehr viel mehr Zeit und vor allem Expertenwissen in verschiedenen Teilgebieten erforderlich gewesen wäre.

Bei der Hardware stellt die Anzahl der zur Verfügung stehenden Sensor-/Aktorschnittstellen an den Bausteinen und Funktionsträgern bei komplexeren Aufbauten ein prinzipielles Problem dar. Bei maximalem Ausbau stehen vier analoge Eingänge (single ended), zwei analoge Ausgänge und zwei digitale (Last-) Ausgänge zur Verfügung. Im Fall des Pneumatikzylinders wäre für einen besser überwachten und geregelten Aufbau Bedarf für einen weiteren Wegsensor, Endlagenschalter und eine Drucküberwachung. Bei komplexeren Sensor-/Aktorsystemen reicht somit die Anzahl dieser Schnittstellen nicht mehr aus.

Das Nichterreichen der Regelgenauigkeit von $\pm 1,0\text{mm}$ liegt in erster Linie am gewählten Systemaufbau. Die verwendeten Ventile weisen für diese Anwendung eine zu geringe Schaltfrequenz auf. Schnell schaltende Piezoventile sind für diesen Aufbau zu bevorzugen, waren aber bei der Durchführung der Arbeiten noch nicht verfügbar.

Parallel konnten mit diesem Versuchsaufbau auch Einschränkungen und Grenzen des Systems EasyLab aufgezeigt werden. Die Einstellmöglichkeiten der vorgegebenen Regler sowie der Unterprogramme hinsichtlich ihres Zeitverhaltens sind noch unzureichend. Durch die Kombination aus ablaufgesteuertem und Datenfluss gesteuertem Programm können einfach alternative Wege zur Programmierung beschritten werden. Doch insbesondere bei zeitkritischen Prozessen, wie es beim Pneumatikzylinder der Fall war, reichen die Einstellmöglichkeiten an den einzelnen Funktionsbibliotheken häufig noch nicht aus. Allerdings ist anzumerken, dass diese beschriebenen Probleme bei einem weiteren Ausbau von EasyLab und entsprechenden Rückmeldungen an die Programmierer beseitigt und das Werkzeug so noch deutlich effektiver gestaltet werden kann.

Ebenso wäre ein leistungsfähiger CPU-Baustein wünschenswert. Dieser befindet sich aber in Entwicklung und sollte bei Fortführung der Arbeiten nutzbar sein.

8 Vergleich Wirtschaftlichkeit am Beispiel des Temperatormesssystems

Neben den technischen Möglichkeiten ist die Wirtschaftlichkeit ein wesentliches Argument, das über den Einsatz von Entwicklungswerkzeugen und der Entwicklungsmethodik in der industriellen Praxis entscheidet. Daher erfolgt am Beispiel des Temperatormesssystems ein Vergleich der Wirtschaftlichkeit der Entwicklungsmethodik EasyKit mit einem konventionellen Vorgehen bei der Entwicklung und der anschließenden Produktion.

Um eine Vergleichbarkeit zu ermöglichen, sind Rahmenbedingungen und Annahmen festzulegen. Folgende Fälle sollen sowohl hinsichtlich der Entwicklungskosten als auch der erforderlichen Entwicklungszeit sowie den anschließenden Produktstückkosten miteinander verglichen werden:

- Entwicklungskosten bei Durchlaufen von 3 Stufen (A-, B-, C-Muster bzw. Stufe 1, 2 und 3) für den Aufbau eines Temperatormesssystems für die in Kapitel 7.1.1 beschriebenen Anforderungen bei:
 - a) Konventionellen Vorgehen
 - b) Entwicklungsmethodik EasyKit bei komplett vorliegendem Bausteinsatz
 - c) Entwicklungsmethodik EasyKit bei einer fehlenden Funktion
- Produktstückkosten eines Temperatormesssystems bei jeweiliger vollständiger Umlage der Entwicklungskosten auf die jeweilige Stückzahl für die bereits verglichenen Vorgehen a) bis c) und für die Stückzahlen 10 Stück / 50 Stück / 500 Stück / 5.000 Stück

Ein solcher Vergleich basiert stets auf Annahmen bei den Kalkulationen (Erfahrungswissen des Bearbeiters, fertigungstechnische Ausrüstung etc.) und kann immer nur ein Anhaltspunkt für einen weiteren Vergleichsprozess sein. Der Vergleich soll dabei so aufbereitet werden, dass es einem Dritten einfach möglich ist, die Daten entsprechend seinen Bedingungen anzupassen. Daher werden die Randparameter im Folgenden detailliert beschrieben und die Tabellen so aufbereitet, dass ein Austausch gegenüber den tatsächlich vorliegenden Verhältnissen einfach möglich ist.

Für die einzelnen Arbeiten wurden industrieübliche Stundensätze angenommen, die je nach Unternehmen oder auch Region für jeden Einzelfall stark variieren können. Für Arbeiten an Maschinen (z.B. Bestückung) wurde auf Basis einer Mischkalkulation ein gemeinsamer Stundensatz (Maschine und Personal) angegeben. Die angegebenen Durchlaufzeiten berücksichtigen die üblichen Materiallaufzeiten (z.B. bei Leiterplatten 10 Arbeitstage) ohne Eilzuschläge. Beim Material werden beim konventionellen Vorgehen die Beschaffungskosten für die einzelnen Komponenten inklusive der Einmalkosten berücksichtigt. Diese Durchlaufzeiten werden für vergleichbare Positionen (z.B. Materialbeschaffung) bei allen zu vergleichenden Varianten als identisch angenommen. Für die Entwicklungsmethodik EasyKit werden die Bausteinkosten zu Grunde gelegt.

Konkret gelten für den nachfolgenden Vergleich die folgenden Annahmen:

- Ingenieurstundensatz Entwicklung (HW + SW): 75,00 EUR / Stunde
- Stundensatz Layouter: 60,00 EUR / Stunde
- Stundensatz Techniker (z.B. Test): 60,00 EUR / Stunde
- Maschinenstundensatz (inkl. Bediener): 80,00 EUR / Stunde

Die normalerweise erforderlichen Untersuchungen für die Freigabe der Systeme für eine Serie (z.B. EMV, Lebensdauer, Vibration) werden in der folgenden Betrachtung nicht berücksichtigt.

8.1 Vergleich der Entwicklungskosten

8.1.1 Entwicklungskosten nach konventionellem Entwicklungsvorgehen

Es wird davon ausgegangen, dass für die Entwicklung erfahrenes Personal (Hard- und Software) zur Verfügung steht und dass dieses in den speziellen Teilgebieten (Netzteil, Signalverarbeitung, Signalaufbereitung) über die notwendigen Erfahrungen verfügt, so dass keine grundlegende Untersuchungen oder Vorarbeiten mehr erforderlich sind. Die vorliegenden Entwicklungswerkzeuge (Schaltplan, Layout) werden beherrscht. Ebenso wird davon ausgegangen, dass nach Eingang der Teile eine sofortige Bestückung möglich ist.

Bei der konventionellen Entwicklung wird für die Erstellung der Firmware davon ausgegangen, dass die Entwicklungsumgebung (z.B. für Microchip die MPLAB

IDE) zur Verfügung steht und der Programmierer in diese eingearbeitet ist und sowohl die Entwicklungsumgebung als auch die Programmiertools beherrscht.

Für den Aufbau des A-Musters wird für den Messteil ein diskreter Aufbau auf Basis eines Evaluationboards mit einhergehenden Funktionstests vorgesehen. Die einzelnen Arbeitsschritte, der dafür erforderliche Zeitaufwand sowie die damit verbundenen Kosten sind in Tabelle 8.1 dargestellt. Somit ergeben sich die in Kosten von 29,6 TEUR und eine Durchlaufzeit von 40 Tagen.

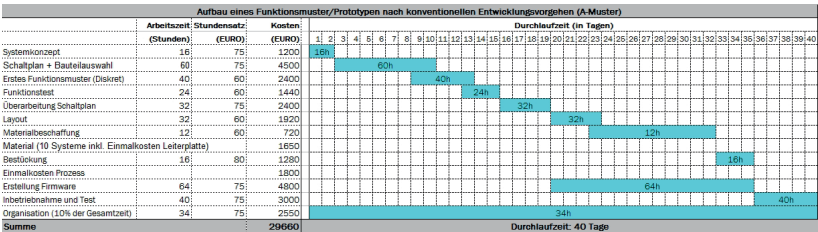


Tabelle 8.1: Entwicklungskosten beim Aufbau eines A-Musters nach konventionellem Vorgehen

Auf Basis der mit dem A-Muster durchgeführten Applikationstests erfolgen erforderliche Anpassungen am Systemkonzept sowie daraus resultierende Anpassungen am Schaltplan und Layout. Es wird davon ausgegangen, dass ca. 80% des A-Musters übernommen werden können und aus technischer Sicht vor allem Anpassungen im Messeingang der Schaltung und für EMV-Maßnahmen erforderlich sind. Ebenso fallen die Einmal- und Bestückungskosten geringer aus, da zum Teil auf vorhandene bzw. anzupassende Bibliotheken zurückgegriffen werden kann. In Summe entstehen für den Aufbau des B-Musters Kosten von 15,9 TEUR bei einer Durchlaufzeit von 30 Tagen.

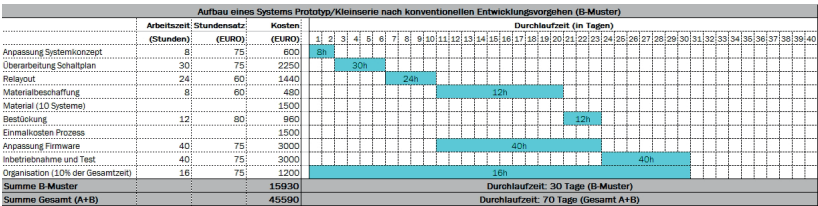


Tabelle 8.2: Entwicklungskosten beim Aufbau eines B-Musters nach konventionellem Vorgehen

Der Schwerpunkt des C-Musters ist eine wirtschaftliche Optimierung. Hierzu werden Bauelemente geprüft, die Anordnung der Bauelemente für eine optimale Bestückung korrigiert und daraus resultierende letzte Anpassungen an der Firmware vorgenommen. Da die Freigabetests in dieser Aufstellung nicht berücksichtigt wurde die Durchlaufzeit für die C-Muster Phase mit 20 Tage ermittelt, die Kosten mit 9,2 TEUR.

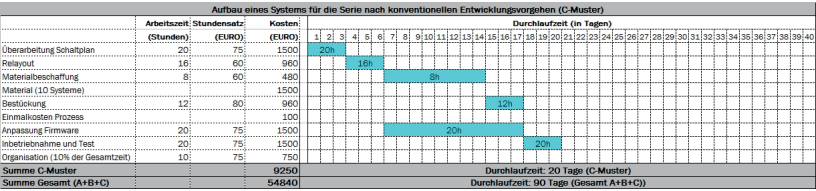


Tabelle 8.3: Entwicklungskosten beim Aufbau eines C-Musters nach konventionellem Vorgehen

8.1.2 Entwicklungskosten der Entwicklungsmethodik EasyKit bei komplett vorliegendem Bausteinsatz

Für den Aufbau der ersten Systeme wird das Vorgehen nach Stufe 1 der Entwicklungsmethodik berücksichtigt, d.h. der Aufbau eines ersten Musters und die Erstellung der Firmware sowie ein erster Funktionstests erfolgen mit den steckbaren Bausteinen. Es wird auch hier davon ausgegangen, dass der Entwickler über grundlegende Erfahrungen über die Entwicklungsmethodik EasyKit verfügt und eine Einarbeitung nicht erforderlich ist. Dies gilt auch für die Werkzeuge in EasyKit, insbesondere das Programmierwerkzeug EasyLab. Somit ergeben sich die in Tabelle 8.4 dargestellten Kosten von 7,2 TEUR und eine Durchlaufzeit von 14 Tagen für den Aufbau eines ersten Musters.

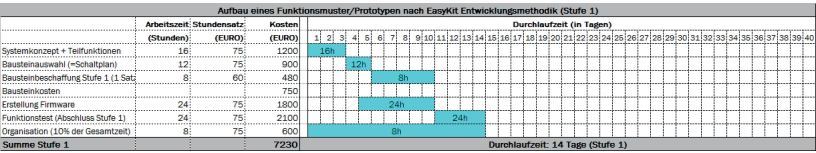


Tabelle 8.4: Entwicklungskosten beim Aufbau der Stufe 1 nach der EasyKit Entwicklungsmethodik

Für den Übergang zu Stufe 2 ist bei der Entwicklungsmethodik EasyKit lediglich die Umstellung von den steckbaren Bausteinen auf die lötbaren Funktionsträger erforderlich. Für die Firmware wird eine Optimierungsschleife vorgesehen sowie

Aufbau eines Funktionsmuster/Prototypen nach EasyKit Entwicklungsmethodik (Stufe 2)			
Arbeitszeit: (Stunden)	Stundenanzahl: (EURO)	Kosten (EURO)	Durchlaufzeit (in Tagen)
Anpassung Systemkonzept	8	75	600
Systementwurf Stufe 2 ILO 9b	8	60	480
Bausteine ILO Satz Stufe 2			2100
Montage und Basisset	16	75	1200
Anpassung Firmware	20		
Funktionstest	16	75	1200
Organisation (10% der Gesamtzeit)	8	75	600
Summe Stufe 2			6190
Summe Gesamt (1+2)			13410

Durchlaufzeit: 15 Tage Stufe 2
Durchlaufzeit: 20 Tage Gesamt (1+2)

Für den Übergang in Stufe 3 ist bei der Entwicklungsmethodik EasyKit die Zusammenführung der einzelnen Teilfunktionen auf Schaltplanebene erforderlich (hier erfolgt die Annahme, dass der Zugriff auf diese direkt möglich ist) sowie die Erstellung eines Layouts auf einen eigenständigen Schaltungsträger. Auf eine wirtschaftliche Optimierung wird verzichtet, um die Firmware aus Stufe 2 ohne größere Änderungen weiter verwenden zu können. Somit fallen für den Aufbau des Temperaturmesssystems nach Stufe 3 der Entwicklungsmethodik EasyKit 13,9 TEUR bei einer Durchlaufzeit von 30 Tagen an.

Aufbau eines Systems für die Serie nach EasyKit Entwicklungsmethodik (Stufe 3)																																																			
	Arbeitszeit:	Stundensatz:	Kosten:	Durchlaufzeit (in Tagen)																																															
	(Stunden)	(EURO)	(EURO)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40								
Überführung Schaltplan	24	75	1800	24h																																															
Layout	40	60	2400													40h																																			
Materialbeschaffung	12	60	720													12h																																			
Material (10 Systeme)																																																			
Bestückung	12	80	960																									12h																							
Einmalkosten Prozess	12		1500																																																
Prüfung Firmware	16	75	1200																																																
Inbetriebnahme und Test	32	75	2400																	16h																															
Organisation (10% der Gesamtzeit)	16	75	1200																																	32h															
Summe			13930																																																
Summe Gesamt (1+2+3)			27340																																																
				Durchlaufzeit: 30 Tage Stufe 3																																															
				Durchlaufzeit: 59 Tage (Gesamt 1+2+3)																																															

8.1.3 Entwicklungskosten nach der Entwicklungsmethodik EasyKit bei einer Bausteinneuentwicklung

Im konkreten Fall des Temperaturmesssystems wird davon ausgegangen, dass die Spannungsaufbereitung und die Signalverarbeitung als stets wiederkehrende Funktionen zur Verfügung stehen, der ADC-Baustein hingegen neu ausgelegt werden muss (z.B. als 16bit ADC mit spezieller Vorverstärkung). Es ist zu berücksichtigen, dass für den neuen Baustein auch entsprechende Funktionen in EasyLab anzulegen sind. Es erfolgt der Aufbau eines einfachen Funktionsmusters, auf dessen Basis unter Berücksichtigung der Schnittstellen an-

Aufbau eines neuen Bausteins für die EasyKit Entwicklungsmethodik			Durchlaufzeit (in Tagen)																																																		
	Arbeitszeit (Stunden)	Stundensatz: (EUR/Std)	Kosten (EUR/Std)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40										
Konzept Teilfunktion	8	75	600																																																		
Schaltplan + Bauteilauswahl	24	75	1800	24h																																																	
Schaltpläneprüfung/Funktionmuster	24	75	1800					10h																																													
Funktionsmodell	8	60	480							8h																																											
Überarbeitung Schaltplan	12	75	900									1,2h																																									
Layout	20	60	1200											2,2h																																							
Materialbeschaffung	8	60	480																					8h																													
Material (LO Systeme)			1000																																																		
Bestückung	12	80	960																																																		
Erstellung Funktionsblock Software	40	75	3000																																			40h		12h													
Inbetriebnahme und Test	20	75	1500																																																		
Organisation (10% der Gesamtzeit)	18	75	1200																																			20h															
Summe			14920	Durchlaufzeit: 34 Tage																																																	

Da diese Entwicklung praktisch in die Systementwicklung für das Temperaturmesssystem eingeschoben werden muss, werden für die Betrachtung der gesamten Kosten für ein System nach Stufe 1 der Entwicklungsmethodik die Kosten und Durchlaufzeiten für den Baustein mit denen für den Aufbau eines Funktionsmusters (siehe Tabelle 8.4) summiert. Somit ergeben sich Gesamtkosten von 22,1 TEUR bei einer Gesamtdurchlaufzeit von ca. 45 Tagen (es wurde eine Überlappung mit der Entwicklung des Gesamtsystems von 10% angenommen).

8.1.4 Zusammenfassung Entwicklungskosten

Der Übergang vom Prototyp in ein serienreifes System ist in beiden Fällen sowohl von den Kosten als auch vom erforderlichen Zeitraum vergleichbar.

Bei erforderlicher Bausteinneuentwicklung geht der Zeitvorteil verloren. Durch den neu zu entwickelnden Baustein ist in Summe von einer längeren Entwicklungsdauer bis zu den ersten vorliegenden Systemen auszugehen. Die Kosten für die Entwicklung sind jedoch bei der Entwicklungsmethodik EasyKit trotzdem geringer. Hier ist davon auszugehen, dass durch die vorliegenden Werkzeuge und Hilfsmittel sowie die Abgrenzung auf einen Funktionsblock weniger Aufwand erforderlich ist und somit die Kosten geringer ausfallen.

Entwicklungsstand	Konventionelle Entwicklung		EasyKit ohne Bausteinneuentwicklung		EasyKit mit Bausteinneuentwicklung	
	Kosten	Dauer in Tagen	Kosten	Dauer in Tagen	Kosten	Dauer in Tagen
A-Muster / Stufe 1	29.660,00 €	40	7.230,00 €	14	22.150,00 €	45
B-Muster / Stufe 2	15.930,00 €	30	6.180,00 €	15	6.180,00 €	15
C-Muster / Stufe 3	9.250,00 €	20	13.930,00 €	30	13.930,00 €	30
SUMME	54.840,00 €	90	27.340,00 €	59	42.260,00 €	90

Tabelle 8.8: Vergleich der Entwicklungskosten und Dauer

Zusammenfassend ist festzustellen, dass bei nur einmaliger Anwendung der Entwicklungsmethodik die Vorteile sich nur dann voll auswirken, wenn für alle Funktionen Bausteine vorliegen. Im Fall einer erforderlichen Bausteinneuentwicklung sind die Kosten zwar immer noch geringer, es ist jedoch mit einem längeren Zeitraum wie bei einer konventionellen Entwicklung zu rechnen.

Wenn die Entwicklungsmethodik jedoch wiederkehrend eingesetzt wird und somit auch die Anzahl der vorhandenen Funktionsbausteine in Hard- und Software wachsen, ergeben sich deutliche Vorteile für das anwendende Unternehmen. Des Weiteren ist vorteilhaft, auf bereits getestete Funktionsgruppen in Hard- und Software zurückgreifen zu können. Dieser Punkt wurde in der Kostenbetrachtung nicht näher berücksichtigt, da der Aufwand zur Fehlersuche in Schaltungen und in der Software nur schwer zu beziffern ist und sich von Fall zu Fall extrem unterscheiden kann.

Ebenso ist in die Bewertung das Vorhalten von Experten nicht eingegangen. Im Fall der konventionellen Entwicklung sind Experten in Elektronik, Software und Messtechnik erforderlich. Bei der Entwicklungsmethodik EasyKit ist davon auszugehen, dass ohne neuen erforderlichen Baustein diese Systementwicklung

auch von einem Messtechniker oder Elektroniker durchgeführt werden kann. Dies bedeutet, dass mit EasyKit auch für ein Unternehmen mit einer kleinen Entwicklungsabteilung, in der nicht jedes Fachgebiet abgedeckt ist, die Entwicklung eines derartigen miniaturisierten Messsystems einfacher möglich wird.

8.2 Vergleich Produktstückkosten

In einem zweiten Schritt erfolgt der Vergleich der Produktstückkosten für eine Serienproduktion. Wie schon eingangs beschrieben sollen dabei die Stückzahlen 10, 50, 500 und 5.000 verglichen werden. Um den Vergleich eindeutig zu gestalten, wird im Folgenden auch der Entwicklungsstand definiert, zu dem die jeweilige Stückzahl produziert wird. Die Entwicklungskosten werden zu 100% auf die definierte Stückzahl umgelegt:

	Konventionell	Easy-Kit
10 Stück	A-Muster	Stufe 1
50 Stück	B-Muster	Stufe 2
500 Stück	C-Muster	Stufe 3
5.000 Stück	C-Muster	Stufe 3

Tabelle 8.9: Aufstellung der zu vergleichenden Varianten

Die Kosten für die erforderlichen Tests und Freigabeuntersuchungen (z.B. EMV, Lebensdauer) werden im Weiteren nicht berücksichtigt, da sie für beide Entwicklungen als identisch angenommen werden.

8.2.1 Produktstückkosten bei konventionellem Entwicklungsvorgehen

Für die Berechnung der Produktstückkosten beim konventionellen Vorgehen werden folgende Positionen berücksichtigt:

Material:

Summe der Kosten aller Bauelemente (Elektronik, Leiterplatte, Stecker, Gehäuse) pro System.

Materialbeschaffung:

Kosten, resultierend aus dem Zeitaufwand für die Beschaffung, Warenannahme und Lagerung umgelegt auf die jeweilige Stückzahl.

Bestückung:

Kosten für die Bestückung (Maschine aufrüsten, Siebdruck, eigentliche Bestückung, Löten) sowie die Sichtkontrolle der bestückten Leiterplatten. Die Einmalkosten für das Einrichten der Maschine, Siebe etc. sind bereits in den Entwicklungskosten enthalten und werden daher in dieser Position nicht mehr berücksichtigt. Die Kosten berechnen sich aus der Bestückung eines Loses (50, 500 oder 5.000 Stück) umgelegt auf die jeweilige Stückzahl.

Programmierung und Funktionstest:

Das vorliegende System wird mit einer Applikationssoftware bespielt und die Grundfunktionen werden getestet.

Entwicklungsumlage:

Kosten der Entwicklung, komplett umgelegt auf die jeweilige Stückzahl. Beim A-Muster wird davon ausgegangen, dass im Rahmen der Entwicklung mit einer Auflage von 10 Stück diese Systeme bereits realisiert sind und keine weitere Fertigungsschleife erforderlich ist.

	A-Muster	B-Muster	C-Muster	C-Muster
Kosten	10 Stück	50 Stück	500 Stück	5.000 Stück
Material		25,00 €	17,00 €	15,50 €
Materialbeschaffung		18,00 €	3,50 €	1,55 €
Bestückung		19,20 €	3,75 €	3,35 €
Programmierung + Funktionstest		4,50 €	1,35 €	0,90 €
Entwicklungsumlage	2.966,00 €	911,80 €	109,68 €	10,97 €
Stückkosten	2.966,00 €	978,50 €	135,28 €	32,27 €

Tabelle 8.10: Produktstückkosten eines Temperatormesssystems bei konventionellem Vorgehen

8.2.2 Produktstückkosten bei EasyKit

Für die Berechnung der Produktstückkosten beim Aufbau nach der Entwicklungsmethodik EasyKit werden folgende Positionen berücksichtigt:

Material:

In Stufe 1 und 2 werden die Kosten für einen Satz Funktionsträger [EFM2008] zugrunde gelegt, bei Stufe 3 dann analog des konventionellen Vorgehens die

Kosten für die Bauelemente pro System. Da keine wirtschaftliche Optimierung stattfand wird mit Mehrkosten bei den Bauelementen von ca. 20% gerechnet.

Materialbeschaffung:

Kosten, resultierend aus dem Zeitaufwand für die Beschaffung, Warenannahme und Lagerung umgelegt auf die jeweilige Stückzahl.

Montage/Bestückung:

In Stufe 2 werden die Kosten für das Verbinden der Bausteine (Funktionsträger) durch Löten berücksichtigt, in Stufe 3 gelten die Kosten für die Bestückung analog dem konventionellen Vorgehen, wobei auch hier mit einem Mehraufwand von ca. 20% aufgrund der fehlenden Optimierung gerechnet wird.

Programmierung und Funktionstest:

Das vorliegende System wird mit einer Applikationssoftware bespielt und die Grundfunktionen werden getestet.

Entwicklungsumlage:

Kosten der Entwicklung, komplett umgelegt auf die jeweilige Stückzahl.

	Stufe 1	Stufe 2	Stufe 3	Stufe 3
Kosten	10 Stück	50 Stück	500 Stück	5.000 Stück
Material	210,00 €	165,00 €	21,25 €	19,38 €
Materialbeschaffung	30,00 €	15,00 €	4,00 €	1,60 €
Montage/Bestückung	10,00 €	10,00 €	4,00 €	3,50 €
Programmierung + Funktionstest	10,00 €	5,00 €	1,50 €	1,00 €
Entwicklungsumlage	723,00 €	268,20 €	54,68 €	5,47 €
Stückkosten	983,00 €	463,20 €	85,43 €	30,94 €

Tabelle 8.11: Produktstückkosten eines Temperatormesssystems auf Basis der EasyKit Entwicklungsmethodik

Für die Berechnung der Produktstückkosten beim Aufbau nach der Entwicklungsmethodik EasyKit mit Bausteinneuentwicklung werden im Vergleich zu Tabelle 8.11 lediglich die Kosten für die Entwicklungsumlage angepasst, alle anderen Position werden als identisch angenommen.

	Stufe 1	Stufe 2	Stufe 3	Stufe 3
Kosten	10 Stück	50 Stück	500 Stück	5.000 Stück
Material	210,00 €	165,00 €	21,00 €	19,38 €
Materialbeschaffung	30,00 €	15,00 €	4,00 €	1,60 €
Montage/Bestückung	10,00 €	10,00 €	4,00 €	3,50 €
Programmierung + Funktionstest	10,00 €	5,00 €	1,50 €	1,00 €
Entwicklungsumlage	2.215,00 €	566,60 €	84,52 €	8,45 €
Stückkosten	2.475,00 €	761,60 €	115,02 €	32,05 €

Tabelle 8.12: Produktstückkosten eines Temperatormesssystems auf Basis der EasyKit Entwicklungsmethodik mit Bausteinneuentwicklung

8.2.3 Vergleich der Produktstückkosten

Beim Vergleich der Produktstückkosten ist festzustellen, dass die Variante EasyKit ohne Bausteinneuentwicklung sowohl bei kleinsten, kleinen als auch bei mittleren Stückzahlen deutliche Vorteile gegenüber der konventionellen Entwicklungsmethodik aufzuweisen hat. Bei 50 Stück (463,20 EUR EasyKit gegenüber 978,50 EUR konventionell) und bei 500 Stück (85,50EUR EasyKit gegenüber 135,28 EUR konventionell) liegt der Vorteil bei deutlich über 50%, bei 5.000 Stück sind die Kosten dann nahezu vergleichbar, bei noch größeren Stückzahlen würde sich dann die Schere zugunsten der konventionellen Entwicklungsmethodik immer weiter öffnen. Auch unter Berücksichtigung einer Bausteinneuentwicklung dominieren die Kostenvorteile auf Seite der Entwicklungsmethodik EasyKit, wenngleich diese Vorteile nicht mehr so dominant sind. Bereits ab ca. 2.500 Systemen sind dann die Produktstückkosten der konventionellen Fertigung günstiger.

Im Ergebnis ist festzustellen, dass insbesondere bei kleinen und mittleren Stückzahlen von 1 bis ca. 2.500 die Entwicklungsmethodik EasyKit deutliche Kostenvorteile aufweist, sowohl ohne als auch mit erforderlicher Bausteinneuentwicklung.

Da insbesondere in kleineren Unternehmen viele Systeme mit einer Jahresproduktionszahl von 100...500 Stück gefertigt werden, bietet sich die Entwicklungsmethodik EasyKit insbesondere für derartige Unternehmen an. Mit einer wachsenden Zahl von Funktionen und Bausteinen können dadurch sowohl im

Entwicklungsbereich als auch in der Produktion Kosten gespart werden. Und unter Berücksichtigung der viel kürzeren Entwicklungszeiten bei vorliegendem Bausteinsatz können auch in Bezug auf die Schnelligkeit, mit der ein Produkt am Markt platziert werden kann, klare Wettbewerbsvorteile erzielt werden.

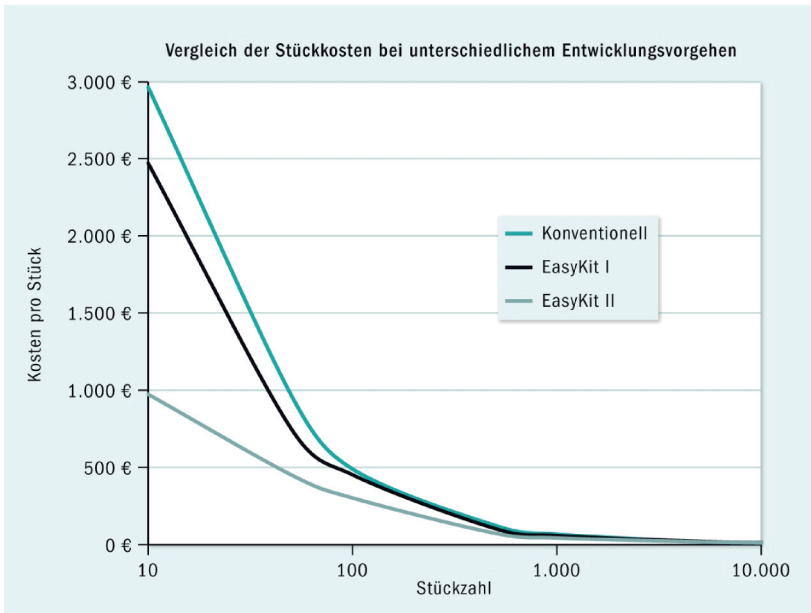


Abbildung 8.1: Vergleich der Entwicklungskosten

Jedoch ist zu berücksichtigen, dass diese Wirtschaftlichkeitsbetrachtung je nach System und Komplexität sehr unterschiedlich ausfallen kann. Insbesondere bei Systemen mit größerer Komplexität (größere Anzahl Schnittstellen, Sensor-/Aktorkombinationen) stehen unter Umständen keine Funktionen in Hard- und Software zur Verfügung und sind nur mit größerem Aufwand als Funktionsblöcke zu entwickeln. In solchen Fällen kann die Entscheidung für die Entwicklungsmethodik zu längeren Entwicklungszeiten und damit verbunden höheren Kosten führen. Daher empfiehlt sich zu Beginn der Entwicklung stets eine Wirtschaftlichkeitsbetrachtung, in der die Verfügbarkeit von Funktionen und der Aufwand zur Entwicklung dieser Funktionen einem konventionellen Entwicklungsvorgehen gegenübergestellt werden.

9 Zusammenfassung

Zielstellung der Arbeit war die Einführung und Erprobung einer speziell für die Anforderung der Entwicklung von miniaturisierten mechatronischen Systemen und von eingebetteten Systemen angepassten Entwicklungsmethodik und der Nachweis der technischen und wirtschaftlichen Machbarkeit. Mit EasyKit und den dafür entwickelten Werkzeugen liegt nunmehr eine solche Entwicklungsmethodik vor, die sich in ihrer Struktur an der Entwicklungsmethodik für mechatronische Systeme VDI2206 und dem damit verbundenen V-Modell orientiert, gleichzeitig aber die spezifischen Anforderungen für den Aufbau miniaturisierter und eingebetteter Systeme berücksichtigt.

Ein besonderes Augenmerk lag auf einem einfachen, fachübergreifenden Zugang zu den Entwicklungswerkzeugen. Durch den Rückgriff auf vorhandene Funktionsblöcke in Hard- und Software sowie die Schaffung begleitender Entwicklungswerkzeuge, die durch kompatible Schnittstellen wiederkehrend eingesetzt werden können, ist dieser Zugang gelungen. Offene Schnittstellen ermöglichen einen freien Zugang zum System. Dabei ist jedoch zu berücksichtigen, dass für die Erweiterung der Funktionen Experten in den einzelnen Teildisziplinen erforderlich sind.

Das Drei-Stufen-Modell bietet die Möglichkeit eines schnellen Aufbaus von Prototypen und eines ebenso schnellen und sicheren Übergangs vom Prototyp zur Kleinserie und anschließend zu einem Serienprodukt. Voraussetzung für die praktische Anwendung des Drei-Stufen-Modells war die Schaffung einer Technologieplattform, die sowohl den schnellen und flexiblen Aufbau von Prototypen als auch auf gleicher Technologiegrundlage den Aufbau von industrietauglichen Systemen in Kleinserie ermöglicht. Mit der hier entwickelten Verbindungstechnologie für die Stufe 1 (Steckverbindung) und für die Stufe 2 (Lötverbindung über außen liegenden Bus) kann dieser Übergang optimal und unter Nutzung der identischen Funktionsblöcke erfolgen. Durch gleichzeitigen Rückgriff auf die Schaltungs- und Layoutbibliotheken dieser fertig entwickelten Funktionsblöcke ist ebenfalls der Übergang zu Stufe 3 einfach und schnell möglich.

Parallel zur Hardware ist für den Erfolg einer derartigen Entwicklungsmethodik eine sichere und effizient zu erstellende Software von grundlegender Bedeutung. Mit EasyLab und den dazugehörigen Funktionsbibliotheken steht nunmehr ein solches Softwarewerkzeug zur Verfügung. Insbesondere der einfache Zugang

für Nicht-Informatiker, speziell durch die grafische Programmieroberfläche, und der wiederkehrende und sichere Einsatz getesteter Funktionsbibliotheken über alle drei Stufen des Entwicklungsprozesses sind positiv hervorzuheben. Durch die Abbildung hardwarespezifischer Funktionsblöcke in EasyLab wird darüber hinaus das Hardware/Software Co-Design unterstützt. Bereits im frühen Entwurfsprozess können die Anforderungen auf Hard- und Software abgebildet und geeignete Plattformen ermittelt werden.

Neben diesen zentralen Punkten spielt auch die Werkzeugkette entlang des Entwicklungsprozesses eine große Rolle. Durch die einheitlichen Schnittstellen in Stufe 1 und 2 der Entwicklungsmethodik konnten für die Entwicklung, die Fertigung und den Test begleitende Vorlagen und Werkzeuge erstellt werden, die die Entwicklung neuer Bausteine und Systeme unterstützen und Zeit und Kosten für den Entwicklungsprozess deutlich senken können.

Neben dem technischen Aspekt ist die Wirtschaftlichkeit entscheidend. Am Beispiel eines konkreten Messsystems konnte der Nachweis der Wirtschaftlichkeit der Entwicklungsmethodik EasyKit erbracht werden. Es ist an dieser Stelle noch einmal zu betonen, dass das Ergebnis dieser Wirtschaftlichkeitsbetrachtung nicht für die Entwicklung miniaturisierter mechatronischer Systeme verallgemeinert werden kann, sondern bei jeder Neuentwicklung ein entsprechender Vergleich zwischen einem Vorgehen nach der Entwicklungsmethodik EasyKit und dem konventionellen Vorgehen sinnvoll ist. Grundlegend ist anzumerken, dass insbesondere bei Sensorsystemen und einfachen Sensor-/Aktorsystemen die Entwicklungsmethodik EasyKit in Verbindung mit den vorliegenden Werkzeugen deutliche Kostenvorteile bringt und damit verbunden auch kürzere Entwicklungszeiten aufweisen kann. Bei komplexeren Systemen ist ein sehr gut und umfassend gefüllter Baukasten aus Hard- und Softwarefunktionen erforderlich. Dies kann nur durch einen konsequenten Einsatz in unterschiedlichsten Bereichen und die damit verbundene Erweiterung, auch über einzelne Firmengrenzen hinaus, erreicht werden.

Der Nachweis der technischen Umsetzbarkeit erfolgte an zwei konkreten Systemaufbauten, einem Temperaturmesssystem und einem positionsgesteuerten Pneumatikzylinder. Für das Temperaturmesssystem erwies sich die Eignung als sehr gut, aus den vorhandenen Bausteinen und Funktionen konnte direkt ein geeignetes System aufgebaut werden. Betrachtungen für vergleichbare Messsysteme für andere Messgrößen zeigen ein vergleichbares Ergebnis. Beim

Pneumatikzylinder und somit beim Einsatz für die Steuer- und Regelungstechnik weist das System in seiner jetzigen Ausführung noch Grenzen auf. Insbesondere die zu geringe Anzahl an Ein- und Ausgängen ist hier limitierend. In allen Fällen ist jedoch die Erstellung der Software positiv hervorzuheben.

Für einen langfristigen Erfolg der Entwicklungsmethodik EasyKit ist ein breiter Einsatz erforderlich. Daher wurden alle Schnittstellen und beschriebenen Werkzeuge offen gelegt und können frei verwendet werden. Eine Standardisierung fand im Rahmen des VDMA Einheitsblatts 66305 statt.

Einrichtungen und Unternehmen, die die Entwicklungsmethodik EasyKit und die Werkzeuge anwenden, können darüber hinaus auch von geprüften Funktionen in Hard- und Software profitieren und so die Entwicklungssicherheit steigern.

Trotz aller Vorteile bedeutet der Einsatz dieses Systems für viele Unternehmen und deren Entwickler einen Paradigmenwechsel. Neben der Einarbeitung in die Methodik und deren Werkzeuge (die in der Regel parallel zu den etablierten Prozessen erfolgt und somit erstmal als Mehraufwand gesehen wird) lebt das System von umfangreichen Funktionsbibliotheken in Hard- und Software. Erst wenn diese in ausreichender Anzahl zur Verfügung stehen, wird der Schritt zu diesem System überschaubar und für die Interessenten leicht. Doch diese Funktionen können nicht von Anfang an zur Verfügung stehen, sondern müssen mit dem System wachsen. Somit bleibt die Frage, ob die Entwicklungsmethodik EasyKit genügend Interessierte findet, um die Funktionsbibliotheken zu füllen und das System zum Erfolg zu führen.

10 Abbildungs- und Tabellenverzeichnis

10.1 Abbildungsverzeichnis

Abbildung 2.1:	Vorgehen zur Ableitung der Entwicklungsmethodik und zur Entwicklung der dafür erforderlichen Werkzeuge	22
Abbildung 3.1:	Erweitertes Phasenmodell der Konstruktionsmethodik	23
Abbildung 3.2:	V-Modell als Makrozyklus nach VDI2206.....	24
Abbildung 3.3:	Sequentieller Entwurfsablauf der Mechatronik.....	25
Abbildung 3.4:	Baukastenkonzept Modulare Mikrosysteme für den Maschinen- und Anlagenbau	28
Abbildung 3.5:	Top-Bottom-BGA.....	30
Abbildung 3.6:	Geometrisch/elektrische Schnittstelle der Bausteingröße 12,5mm x 12,5mm.....	31
Abbildung 3.7:	Elektrische Schnittstelle (Anschlussbelegung)	32
Abbildung 3.8:	Links: Bausteine der ersten Generation (LTCC-Technologie), Mitte: Bausteine der zweiten Generation (FR4-Technologie mit geklebtem Deckel), rechts: Bausteine der dritten Generation (FR4-Technologie mit gelötetem Deckel).....	34
Abbildung 3.9:	Entwurfsalgorithmus des Baukastensystems Modulare Mikrosysteme	35
Abbildung 4.1:	Entwicklungsmethodik und –werkzeuge.....	38
Abbildung 4.2:	Entwicklungsablauf	41
Abbildung 4.3:	V-Modell unter Nutzung der EasyKit Entwicklungsmethodik.....	43
Abbildung 4.4:	Angepasster Entwurfsalgorithmus für die Entwicklungsmethodik EasyKit unter Berücksichtigung des V-Modells.....	44
Abbildung 5.1:	Aufbauform 1 – Funktionsträger für steckbare Rahmen	48
Abbildung 5.2:	Aufbauform steckbarer Kunststoffrahmen	49
Abbildung 5.3:	Bausteine mit Kunststoffrahmen.....	50

Abbildung 5.4:	Konzept mit außen liegendem Bus	53
Abbildung 5.5:	Nutzen mit Funktionsträgern.....	54
Abbildung 5.6:	Vom Funktionsträger (links) über ein Modul mit Rahmen- elementen (Mitte) zum kompletten Baustein (rechts).....	54
Abbildung 5.7:	Thermische Simulation zur Wärmeverteilung in Bausteinen (Quelle: Amic GmbH).....	55
Abbildung 5.8:	Links: Layout eines Funktionsträgers in Starr-Flex-Tech- nologie, rechts: Bestückter Funktionsträger vor dem Falten	56
Abbildung 5.9:	Links: Funktionsträger, Mitte: Bestückter und gefalteter Baustein, rechts: Kompletter Baustein im Gehäuse	57
Abbildung 5.10:	Links – Grid-Array als Schnittstelle zwischen den Bausteinen, rechts – zu verbindende Bausteine	60
Abbildung 5.11:	Elemente der Federmatrix	61
Abbildung 5.12:	Tubus mit Bausteinen und Federmatrizen.....	62
Abbildung 5.13:	Kontaktfolie (12,5mm x 12,5mm) nach Benutzung mit sichtbaren Spuren der BGA.....	64
Abbildung 5.14:	Testboard mit Kontaktfolie	64
Abbildung 5.15:	Matrixstecker im 8x8 Raster mit Orientierungshilfe auf Deckel- und Bodenplatine in Aufbauvariante 2 und 3	66
Abbildung 5.16:	Bausteinstapel über Matrixstecker verbunden	66
Abbildung 5.17:	EasyKit Baugruppe bestehend aus Funktionsträger mit Rahmenelementen	71
Abbildung 5.18:	Systemaufbau in Stufe 2	73
Abbildung 5.19:	EasyKit Baugruppe auf Leiterplatte bestückt	73
Abbildung 6.1:	Erweiterte Anschlussbelegung der elektrischen Schnitt- stelle am Beispiel eines Bausteins der Größe 12,5mm.....	76
Abbildung 6.2:	Belegung der elektromechanischen Schnittstelle.....	77
Abbildung 6.3:	Verschiedene Ausführungen von Funktionsträgern.....	78

Abbildung 6.4:	Komponenten (v.l.n.r.: Deckel/Boden, Rahmen, Interfaces)	79
Abbildung 6.5:	Bausteine (Links mit Grid-Array, Rechts mit Federmatrix)	80
Abbildung 6.6:	Teilfunktionen des Beschleunigungsmesssystems	83
Abbildung 6.7:	Datenblatt des Funktionsblocks Spannungsaufbereitung der EasyKit Entwicklungsmethodik	84
Abbildung 6.8:	Blockschaltbild der Funktion Spannungsaufbereitung	84
Abbildung 6.9:	Schaltplan des Beschleunigungsmessmoduls	86
Abbildung 6.10:	Layoutvorlage für einen Funktionsträger	87
Abbildung 6.11:	Layout Funktionsmodul Beschleunigungssensor	88
Abbildung 6.12:	Layout eines Nutzens (links ohne Kantenfräsung, rechts mit Kantenfräsung)	88
Abbildung 6.13:	Manuell teilbestückte Leiterplatte im Nutzen	89
Abbildung 6.14:	Maschinell bestückter Nutzen	90
Abbildung 6.15:	Erforderliche Komponenten zur Komplettierung des Bausteins (links) und Fertigungshilfsadapter mit Baustein (rechts)	91
Abbildung 6.16:	Gesäuberte Kontaktstellen vor dem Löten (links), Löt- prozess (Mitte) und gelötete Seite eines Bausteins (rechts)	91
Abbildung 6.17:	Funktionsgruppen des Beschleunigungsmesssystems in Stufe 2	92
Abbildung 6.18:	Beschleunigungsmesssystem im Edelstahlgehäuse mit Durchmesser von 20mm	93
Abbildung 6.19:	Datenblatt Funktionsblock Beschleunigungssensor der EasyKit Entwicklungsmethodik	94
Abbildung 6.20:	Umverdrahtungsadapter mit Sockel (links) und eingebunden in Testaufbau (rechts)	95
Abbildung 6.21:	Testadapter (links) und mit eingesetztem Baustein (rechts)	96
Abbildung 6.22:	Testboard	97

Abbildung 6.23: Systemanordnung mit 2 Bausteinen im Testboardverbund	97
Abbildung 6.24: Grafische Programmieroberfläche von EasyLab – Hauptprogramm mit Ablaufschema	100
Abbildung 6.25: Grafische Programmieroberfläche von EasyLab – Unterprogramm mit Datenflussschema	101
Abbildung 6.26: Hardwareunabhängige und hardwarespezifische Funktionsbibliotheken	102
Abbildung 6.27: Menü Codegenerierung	104
Abbildung 7.1: Funktionen des Temperaturmesssystems.....	107
Abbildung 7.2: Eingangsbeschaltung des ADC-Bausteins mit 24bit AD- Wandler.....	108
Abbildung 7.3: Zuweisung der EasyKit Funktionsblöcke zu den erforderlichen Teilfunktionen.....	110
Abbildung 7.4: Pt100 Messsystem aus EasyKit Bausteinen mit Anschluss- set (links: Prinzipaufbau, rechts: realisiertes System).....	110
Abbildung 7.5: Ablaufschema der Firmware für das Temperaturmesssystem.....	111
Abbildung 7.6: Visualisierungssoftware	113
Abbildung 7.7: Modbus Funktion und Detailbeschreibung.....	116
Abbildung 7.8: Umrechnung des ADC Rohwertes in die Spannung und Widerstand über den Pt100	117
Abbildung 7.9: EasyLab Programm für Temperatursensorsystem.....	118
Abbildung 7.10: Vollständiges Programm zur Messwertberechnung der Temperatur in EasyLab.....	119
Abbildung 7.11: Links – Bausteine und Komponenten für Systemaufbau, Mitte - Montiertes Gesamtsystem, rechts – Komplettes Temperaturmesssystem mit Gehäuse.....	122
Abbildung 7.12: Links – Funktionsträger, rechts – Temperaturmesssystem nach Stufe 2 der Entwicklungsmethodik.....	123

Abbildung 7.13: Elektronik des Temperaturmesssystems nach Stufe 3 der Entwicklungsmethodik.....	125
Abbildung 7.14: Technisches Modell.....	126
Abbildung 7.15: Blockschaltbild der Strecke mit Stelleinrichtung.....	127
Abbildung 7.16: Blockschaltbild des Regelkreises.....	128
Abbildung 7.17: Sensor-/Aktorkette	130
Abbildung 7.18: Blockschaltbild der Zuordnung der Bausteine zum geplanten Systemaufbau/ Regelkreis.....	134
Abbildung 7.19: Lösung 1 des Anschlussinterface: Links und Mitte Starr-Flex-Leiterplatte, rechts Leiterplatte mit Bausteinstapel integriert in Abschlussdeckel	136
Abbildung 7.20: Lösung 2 des Anschlussinterface: links Grundaufbau, rechts Aufbau mit Bausteinstapel	136
Abbildung 7.21: Pneumatikzylinder mit Anschlussinterface 1.....	138
Abbildung 7.22: Pneumatikzylinder mit Anschlussinterface 2.....	138
Abbildung 7.23: Teilfunktion für die Berechnung der Position des Pneumatikzylinders.....	140
Abbildung 7.24: Signalstabilität des Messwertes in EasyLab mit und ohne Glättungsfunktion.....	141
Abbildung 7.25: Funktionsblöcke in EasyLab zur Verwendung für den Pneumatikzylinder.....	141
Abbildung 7.26: Zweipunktregler mit Toleranzgrenzen und Ventilen als Stellglieder in EasyLab.....	142
Abbildung 7.27: Plot des Anfahrens einer Position mittels Zweipunktregler	143
Abbildung 7.28: Ablaufprogramm und Unterprogramm 3 in EasyLab	144
Abbildung 7.29: Plot des Anfahrens der Zielposition mit Dämpfung.....	145
Abbildung 8.1: Vergleich der Entwicklungskosten	158

10.2 Tabellenverzeichnis

Tabelle 5.1:	Gegenüberstellung der Aufbauformen	59
Tabelle 5.2:	Kostenpositionen für Berechnung der Systemkosten	69
Tabelle 5.3:	Vergleich der Kosten der Aufbauvarianten	70
Tabelle 5.4:	Gegenüberstellung der Verbindungsarten	70
Tabelle 6.1:	Vergleich der Eignung verschiedener Adapter für unterschiedliche Aufgaben	98
Tabelle 7.1:	Aufstellung der Teilfunktionen	107
Tabelle 7.2:	Genauigkeit des Systems mit konventionell erstellter Firmware	115
Tabelle 7.3:	Genauigkeit des Systems mit EasyLab Firmware.....	118
Tabelle 7.4:	Vergleich der Softwarewerkzeuge.....	121
Tabelle 7.5:	Aufstellung der Teilfunktionen des Pneumatikzylinders.....	129
Tabelle 7.6:	Komponenten und Anschlussarten	135
Tabelle 7.7:	Hardwareabhängige Funktionsblöcke in EasyLab zur Erstellung der Firmware für den Pneumatikzylinder.....	139
Tabelle 8.1:	Entwicklungskosten beim Aufbau eines A-Musters nach konventionellem Vorgehen	149
Tabelle 8.2:	Entwicklungskosten beim Aufbau eines B-Musters nach konventionellem Vorgehen	149
Tabelle 8.3:	Entwicklungskosten beim Aufbau eines C-Musters nach konventionellem Vorgehen	150
Tabelle 8.4:	Entwicklungskosten beim Aufbau der Stufe 1 nach der EasyKit Entwicklungsmethodik	150
Tabelle 8.5:	Entwicklungskosten beim Aufbau der Stufe 2 nach der EasyKit Entwicklungsmethodik	151
Tabelle 8.6:	Entwicklungskosten beim Aufbau der Stufe 3 nach der EasyKit Entwicklungsmethodik	151

Tabelle 8.7:	Entwicklungskosten beim Aufbau eines Funktionsmusters nach der EasyKit Entwicklungsmethodik und einer fehlenden Teilfunktion	152
Tabelle 8.8:	Vergleich der Entwicklungskosten und Dauer	153
Tabelle 8.9:	Aufstellung der zu vergleichenden Varianten	154
Tabelle 8.10:	Produktstückkosten eines Temperaturmesssystems bei konventionellem Vorgehens	155
Tabelle 8.11:	Produktstückkosten eines Temperaturmesssystems auf Basis der EasyKit Entwicklungsmethodik	156
Tabelle 8.12:	Produktstückkosten eines Temperaturmesssystems auf Basis der EasyKit Entwicklungsmethodik mit Bausteinneuentwicklung	157

11 Literaturverzeichnis

- AMIRI JAM 1998 Amiri Jam, K.; Schindler-Saefkow, F.; Großer, V.; Reichl, H.:
Test und Integration von MEMS in fluidisch-elektrischen
3D-Packages aus FR4
In: VDE-Verlag, Berlin 2004, Fachtagung Elektronische
Baugruppen, ISBN: 3-8007-2813-3, S. 385-392
- AMIRI JAM 2000 Amiri Jam, K.; Hillmann, V.; Braun, T.; Becker, K.-F.;
Ansorge, F.; Großer, V.; Reichl, H.:
Mechatronic solutions for modular MEMS
In: VDE-Verlag, Berlin 2000, VDE World Microtechnologies
Congress, ISBN: 3-8007-2579-7
- AMIRI JAM 2004 Amiri Jam, K.; Leutenbauer, R.; Großer, V.; Simsek, A.;
Reichl, H.:
The Design methodology of 3D-HDI
In: VDE-Verlag, Berlin 1998, Konferenz Micro System
Technologies '98, ISBN: 3-8007-2421-9, pp.595-600
- ANALOG 2008_ADXL Analog Devices Inc.
Datenblatt ADXL202E
www.analog.com, Stand April 2008
- BARNER 2008 Barner, S.; Geisinger, M.; Buckl, C.; Knoll, A.:
EasyLab: Model-based Develop-ment of Software for
Mechatronic Systems.
In: IEEE/ASME International Conference on Mechatronic
and Embedded Systems and Applications, S. 540-545,
2008.
- BARNER 2010 Barner, S.; Geisinger, M.; Huang, J.; Knoll, A.; Bönicke, H.;
Ament, C.; Mades, J.; Pittschellis, R.; Bauer, G.:
EasyKit – Eine allgemeine Methodik für die Entwicklung
von Steuerungskomponenten
In 7. Paderborner Workshop "Entwurf mechatronischer
Systeme", HNI-Verlagsschriftenreihe, Band 272

- BAUER 2001A Bauer, G.; Schünemann, M.; Großer, V.; Leutenbauer, R.;
Amiri Jam, K.; Schäfer, W.:
Match-X, a Fabrication Framework for Modular Micro-
systems.
In: IMAPS (Hrsg.): European System Packaging Workshop.
IMAPS 2001, Barcelona, Reston, o. Zählung
- BAUER 2001B Bauer, G.; Schuenemann, M.; Schaefer, W.; Großer, V.;
Leutenbauer, R.; Amiri Jam, K.:
Modularized Microelectromechanical Systems.
In : Microtec 2000 : Applications, Trends, Visions. VDE
World Microtechnologies Congress., Berlin (DE): VDE,
2000, pp. 537 – 542
- BAUER 2004A Bauer, G.:
Modulares Erfolgsprinzip.
In: Carl Hanser Verlag GmbH & Co. KG (Hrsg.): Mikropro-
duktion, 2/2004, S. 48/49
- BAUER 2004B Bauer, G.; Doslik, P.:
Miniaturisierte Systeme aus dem Baukasten.
In: Carl Hanser Verlag GmbH & Co. KG (Hrsg.): Mechatro-
nik, 8-9/2004, S. 25-27
- BAUER 2006 Bauer, G.:
Wissenschaftliche Entwicklung und Fertigung miniaturi-
sierter Systeme – Match-X Konzept.
In: Clusterland Oberösterreich GmbH (Hrsg.): Internationa-
les Forum Mechatronik 2006, Linz, Tagungsband.
ISBN-10: 3-9502270-0-8
- BAUER 2008 Bauer, G.; Knoll, A.; Barner, S.; Pittschellis, R.:
Neuartige mechatronische Entwicklungsmethodik in Aus-
bildung und Industrie.
In: Projektträger Forschungszentrum Karlsruhe (PTKA)
(Hrsg.): Karlsruher Arbeitsgespräche Produktionsfor-
schung 2008.
Wissenschaftliche Berichte FZKA-PFT 214, ISSN 0948-
1427

- BAUER 2009A Bauer, G.:
EasyKit – Der schnelle Weg zu modularen Mikrosystemen.
In: Arbeitsgemeinschaft Modulare Mikrosysteme (Hrsg.):
Frankfurt 2009, 2. Auflage
- BAUER 2009B Bauer, G.:
3D Packaging Technologies for PCB.
In: VDI/VDE-IT (Hrsg.): mst news.
No. 3/09, June 2009, p.22-23
- BIERHALS 1999 Bierhals; Cuhls; Hüntrup; Schünemann; Thies; Weule:
Mikrosystemtechnik – Wann kommt der Durchbruch?,
Miniaturisierungsstrategien im Technologiewettbewerb
zwischen USA, Japan und Deutschland
Physica-Verlag 1999, ISBN 978-3-7908-1250-3
- BÖNICKE 2010 Bönicke, H.; Ament, C.:
Application of the “EasyKit” development method for control of mechatronical systems of small and medium complexity.
In: Control Applications (CCA), 2010 IEEE International Conference on, p. 316 – 321, Yokohama, Japan, ISBN: 978-1-4244-5362-7
- DZIENGEL 2012 Dziengel, N.; Ziegert, M.; Adler, S.; Kasmi, Z.; Seiffert, M.; Schiller, J.:
Verteilte Ereigniserkennung in drahtlosen Sensornetzen überwacht Zaun-, Reha- und Sportanwendungen.
In: Fachzeitschrift für Aufbau- und Verbindungstechnik Elektronik PLUS, März 2012 Band 14, S. 629-635
- EAGLE 2008 Cadsoft Computer GmbH
Download unter: <http://www.cadsoft.de/download-eagle/?language=de>
Stand Januar 2008
- EFM 2008 efm-systems GmbH
Produktkatalog Bausteine und Komponenten
www.efm-systems.de , Stand Juni 2008

- FESTO 2012_ADN Festo GmbH & Co KG
Datenblatt Kompaktzylinder AND-12-50, Artikelnummer
536203
www.festo.com, Stand Juni 2012
- FESTO 2012_MHA Festo GmbH & Co KG
Datenblatt Magentventil MH1, Miniatur
www.festo.com, Stand Juni 2012
- FESTO 2012_SMAT Festo GmbH & Co KG
Datenblatt Positionstransmitter SMAT-8E, für T-Nut
www.festo.com, Stand Juni 2012
- GEISINGER 2009 Geisinger, M.; Barner, S.; Wojtczyk, M.; Knoll, A.:
A software architecture for model-based programming of
robot systems.
In: Torsten Kröger und Friedrich M. Wahl (Hrsg.), Advances in Robotics Research - Theory, Implementation, Application, p. 135-146, Braunschweig, Juni 2009. Springer-Verlag Berlin Heidelberg.
- GRIMME 1996 Grimme, R.; Schünemann, M.; Großer, V.; Reichl, H.;
Schäfer, W.; Schwaab, G.:
Entwicklungstrends in der industriellen Produktion von
Mikrosystemen.
In: Gens, W. (Hrsg.): Wandel im Maschinenbau durch
Feinwerktechnik und Mikrosystemtechnik. Proc. 41. IWK.
Vol. 1. Ilmenau: TU Ilmenau, 1996, S. 271-276
- GROßER 1999 Großer, V.; Schünemann, M.; Michel, B.; Reichl, H.:
Modular Solutions to Packaging MEMS Devices.
In: SEMI [Hrsg.]: Proc. Int. Conf. on MST/MEMS and the 6
th Eur. Strategic Round Table on Microsystems, April 13,
Munich. Mountain View CA (US): SEMI, 1999, getr. Zähl.
- GROßER 2003 Großer, V.; Vogel, J.:
Match-X - A Methodology or Contruction Kit for Modular
Microsystems.
In: Michel, B. ; Fraunhofer-Institut für Zuverlässigkeit und
Mikrointegration -IZM-, Berlin: MicroCar 2003, micro ma-

- materials, nano materials for automotives. Volume of abstracts : Leipzig (D), June 26, 2003
- HANSEN 1955 Hansen, Friedrich:
Konstruktionssystematik ; eine Arbeitsweise für fortschrittliche Konstrukteure;
Verlag Technik, Berlin, 1955; 2. Aufl.
- HOHWIELER 2012 Hohwieler, Eckhard u.a.:
Selbstorganisierende Produktion – SOPRO
In: Fraunhofer Gesellschaft, Gemeinsamer Abschlussbericht des wissenschaftlichen Vorprojekts, Berlin 2012
- HÖHN 2008 Höhn, Reinhard
Das V-Modell XT. Grundlagen, Methodik und Anwendungen
In: Springer Verlag Berlin, 2008, ISBN 978-3540302490
- IEC 61131-3 DIN IEC 61131-3:2009-12
Speicherprogrammierbare Steuerungen- Teil 3: Programmiersprachen.
In: Beuth Verlag, 2009
- ISERMANN 1999 Isermann, R.:
Mechatronische Systeme – Grundlagen.
Springer Verlag, Berlin et al., 1999
- IRRGANG 2004 Irrgang, K.; Michalowsky, L.:
Temperaturmesspraxis mit Widerstandsthermometern und Thermoelementen.
Vulkan Verlag, Essen, 2004, ISBN 3-8027-2200-0
- JANSCHKE 2011 Janschek, K.:
Systementwurf mechatronischer Systeme: Methoden - Modelle - Konzepte
Springer Verlag Berlin 2010, ISBN 978-3540788768
- KALLENBACH 2001 Kallenbach, E.; Zöppig, V.; Birli, O.; Feindt, K.; Ströhla, T.; Saffert, E.; Schmidt, J.:
Integration mechatronischer Systeme.

4. VDI-Mechatronik Tagung 2001: Innovative Produktentwicklung, Frankenthal, Tagungsband, S. 57 – 72, 2001
- KÜHNL 2010 Kühnl, C.:
Einfach wettbewerbsfähiger durch mechatronische Konzepte.
In: etz Elektrotechnik und Automation, VDE-Verlag, Heft 7/2010
- KRAUSE 2004 Krause, W.:
Konstruktionselemente der Feinmechanik.
Carl Hanser Verlag GmbH & CO. KG; Auflage: 3., aktualisierte Auflage (8. April 2004), ISBN 3446223363
- LEUTENBAUER 1997A Leutenbauer, R.; Großer, V.; Reichl, H.; Michel, B.:
Thermomechanical aspects of modular vertical integration technique (TB2GA).
In: Micro Materials. Micro Mat '97. Proceedings : April 16 - 18, 1997, Berlin, Germany, ISBN: 3-932434-05-6, S.1169-1171
- LEUTENBAUER 1997B Leutenbauer, R.; Großer, V.; Reichl, H.:
Die Entwicklung eines stapelbaren Top-Bottom BGA (TB-BGA)
In: Eder, A.; Reichl, H. [Hrsg.]: SMT/ES&S/Hybrid '97 - Tagungsband : 11. Internationale Messe und Kongress für Systemintegration, 22.-24. April 1997, Nürnberg. Stuttgart: Mesago, 1997, S.77-84
- LEUTENBAUER 1998 Leutenbauer, R.; Großer, V.; Schünemann, M.; Reichl, H.:
The Development of a Top-Bottom-BGA (TB-BGA).
In: Reichl, H.; Obermeier, E. (Hrsg.): Micro System Technologies 98. Proc. 6th Int. Conf. on Micro Electro, opto Mechanical Systems an Components, Berlin: VDE-Verlag, 1998, S. 583-588
- MODBUS 2006 Modbus Organisation
MODBUS Application Protocol Specification V1.1b
www.modbus.org, Dezember 2006

- MOSER 2007 Moser, Christian:
Entwurfsoptimierung von intelligenten mechatronischen Systemen dank grafischen Werkzeugen
In: Institut für ZHAW Mechatronische Systeme IMS (Hrsg.): Internationales Forum Mechatronik 2007, Winterthur, Tagungsband. ISBN 978-3-03-301192-2
- PAHL 2005 Pahl, G.; Beitz, W.:
Konstruktionslehre, Grundlagen erfolgreicher Produktentwicklung, Methoden und Anwendung,
Springer Verlag, Berlin, 2006, 7.Auflage, Kapitel 5 bis 8, ISBN 3-540-34060-2
- PITTSCHELLIS 2010 Pittschellis, R.; Bauer, G.; Geisinger, M.; Barner, S.; Bönicke, H.; Ament, C.; Knoll, A.; Mades, J.; Eberbach, R.; Zimmer, K.; Willimsky, E.:
EasyKit – Innovative Entwicklungs- und Didaktikwerkzeuge für mechatronische Systeme
In: Festo Didactic GmbH & Co KG (Hrsg.), Abschlussbericht des BMBF-Förderprojekts EasyKit, 09/2010
- SAMTEC 2007 Produktkatalog Ausgabe 2007, Serie YFT/YFS, S. 70/71 August 2007
- SCHÄFER 2009 Schäfer, W.; Eckardt, T.; Henke, C.; Kaiser, L.; Kerstan, T.; Rieke, J.; Tichy, M.
Der Softwareentwurf im Entwicklungsprozess mechatronischer Systeme.
In: 7. Paderborner Workshop "Entwurf mechatronischer Systeme", 2010, HNI-Verlagsschriftenreihe, Band 272
- SCHAUMONT 2010 Schaumont, Patrick R.:
A Practical Introduction to Hardware / Software Codesign.
In: Springer Verlag New York, London, Berlin, 2010 ISBN: 978-1441959997
- SCHINDLER 2005 Schindler-Saefkow, F.; Schramm, H.; Amiri Jam, K.; Gaßmann, S.:
3D-PCB-Packaging Technology for Electric and Fluidic Applications

- In: International Conference on Micro Electro, Opto, Mechanical Systems & Components, Proceedings, 2005, München
- SCHMITT 2009 Schmitt, R.; Rauchenberger, J.; Spiegelberger, B.; Stetter, R.: „Reifezeugnis“ für die mechatronische Produktion
In: publish-industry Verlag München, A&D-Kompodium 2008/2009, S.73ff
- SCHÜNEMANN 1996 Schünemann, M.; Schäfer, W.; Schließer, J.; Gehringer, H.: Innovationspotentiale im Maschinenbau durch die Mikrosystemtechnik.
In: Gens, W. (Hrsg.): Wandel im Maschinenbau durch Feinwerktechnik und Mikrosystemtechnik. Proc. 41. IWK. Vol. 1. Ilmenau: TU Ilmenau, 1996, S. 265-270
- SCHÜNEMANN 1998A Schünemann, M.; Großer, V.; Leutenbauer, R.; Bauer, G.; Schäfer, W.; Reichl, H.:
A highly flexible design and production framework for modularized microelectromechanical systems.
In: Proc. IEEE 11th Workshop on Micro Electro Mechanical Systems. Piscataway, NJ (US): IEEE, 1998, S. 597-602
- SCHÜNEMANN 1998B Schünemann, M.; Großer, V.; Leutenbauer, R.; Bauer, G.; Schäfer, W.; Amiri-Jam, K.:
Modulare Mikrosystemtechnik – Bericht für den Anwender.
In: VDMA-Verlag (Hrsg.): VDMA-Berichte.
ISBN 3-8163-0374-9
- SCHÜNEMANN 1999 Schuenemann, M.; Huentrup, V., Bierhals, R.:
Economic Potentials of Miniaturization from the Industrial Viewpoint.
In: mst news (1999), Nr.2, pp. 34-36
- SCHÜNEMANN 2001 Schünemann, M.; Bauer, G.; Stock, A.; Schäfer, W.; Großer, V.; Amiri Jam, K.; Leutenbauer, R.; Reichl, H.:
Modularization of MEMS.
In: Reichl, H. (Hrsg.): Microsystems Technologies 2001, 7th International Conference on Micro, Electro, Opto Mechanical Systems and Components

- Düsseldorf 2001, Berlin, Offenbach: VDE, 2001, S.601-606
- SCHÜNEMANN 2003 Schünemann, M.:
Ein modulares Baukastensystem für anwendungsspezifische Mikrosysteme
Jost Jetter Verlag, Heimsheim 2003, ISBN 3-936947-20-1
- STEINHILPER 2008 Steinhilper, W; u.a.:
Konstruktionselemente des Maschinenbaus
In: Springer Verlag, Berlin 2008, ISBN 978-3540766469
- THIELEMANN 2010 Thielemann, Frank:
Produktentwicklung komplexer technischer Systeme
In: Projektträger Forschungszentrum Karlsruhe (PTKA) (Hrsg.): Karlsruher Arbeitsgespräche Produktionsforschung 2010.
Wissenschaftliche Berichte KIT-PTKA-PFT1,
ISSN 1869-9006
- VdI2206 2003 VDI (Hrsg.):
Entwicklungsmethodik für mechatronische Systeme. VDI-Richtlinie 2206
Beuth Verlag, Berlin 2003
- VdI2221 1993 VDI (Hrsg.):
Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte. VDI-Richtlinie 2221
Beuth Verlag, Berlin, 1993
- VDMA 1998 VDMA (Hrsg.):
Modulare Mikrosystemtechnik, Bericht für den Anwender. Ausgabe 1998,
Beuth Verlag, Frankfurt, ISBN 3-8163-0374-9
- VDMA 2003 VDMA (Hrsg.):
Bausteine und Schnittstellen der Mikrotechnik. Ausgabe 2003-07, Beuth Verlag, Frankfurt
- VAN DE VENN 2006 Van de Venn, W.; Eberst, Ch.:
Technologieplattform Manufuture – Mechatronik als

- Schlüsselkompetenz für die Wettbewerbsfähigkeit der produzierenden Industrie
In: Internationales Forum Mechatronik 2006. Tagungsband.
Linz, S.311-331, ISBN-10: 3-9502270-0-8
- WESTKÄMPER 1997 Westkämper, E.; Schünemann, M.; Schlenker, D.; Grimme, R.; Schäfer, W.:
Industrialisierung der Fertigungstechnik für Mikrosysteme.
In: Micro Engineering 97. Proc. Kongreß und Ausstellung für Mikrosysteme und Präzisionstechnik.
Stuttgart: Stuttgarter Messe- & Kongreßgesellschaft, 1997, S. 41-49
- ZEMAN 2006 Zeman, K.; Hehenberger, P.; Scheidl, R.:
Perfekte Produkte durch Mechatronisierung von Prozessen.
In: Internationales Forum Mechatronik 2006. Tagungsband.
Linz, S.311-331, ISBN-10: 3-9502270-0-8
- ZÖPPIG 2005 Zöppig, V.; Kallenbach, E.; Schmidt, J.:
Entwurf volumenintegrierter netzwerkfähiger Antriebssysteme mit modularer Mikrosystemtechnik. Match-X in der Antriebstechnik. Mechatronik 2005, Innovative Produktentwicklung. Tagung Wiesloch. VDI-Berichte Nr. 1892.1. Düsseldorf: VDI-Verlag
- ZHAO 2012 Zhao, X.; Keutel, T.; Baldauf, M.; Kanoun, O.:
Energy Harvesting for Overhead Power Line Monitoring
In: IEEE 9th International Multi-Conference on Systems, Signals and Devices, 2012, Chemnitz, ISBN: 9783981476644.

